

AD-A121 985

RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME
II SECTION I EVENTS GENERATOR PROGRAMS(U) DECISION
SYSTEMS DAYTON OH W 5 DEMMY MAY 80 TR-80-02-A
F33600-78-C-0524

1/1

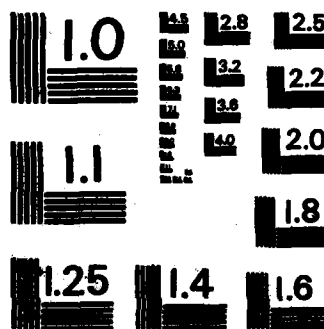
UNCLASSIFIED

F/G 9/2

NL

END

FILED
X
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PHOTOGRAPH THIS SHEET

AD A121985

DTIC ACCESSION NUMBER



LEVEL



INVENTORY

RIME: Vol. II, Section I

DOCUMENT IDENTIFICATION

Rept. No. TR-80-02-A

Contract F33600-78-C-0524 May 80

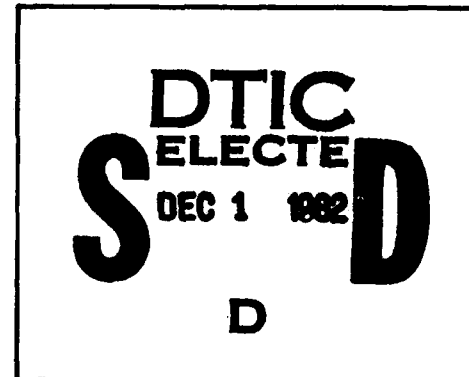
DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR	
NTIS	GRA&I
DTIC	TAB
UNANNOUNCED	
JUSTIFICATION	
BY	
DISTRIBUTION /	
AVAILABILITY CODES	
DIST	AVAIL AND/OR SPECIAL
A	

DISTRIBUTION STAMP



DATE ACCESSIONED



DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

RIME:
The Recoverable Item Management Evaluator:
Volume II, Section I
Events Generator Programs

RIME:
The Recoverable Item Management Evaluator:
Volume II, Section I
Events Generator Programs

by
W. Steven Demmy

May 1980

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

TR-80-02-A

DECISION SYSTEMS
2125 Crystal Marie Drive
Dayton, Ohio 45431

(513) 426-8515

82 11 30 025

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RIME: The Recoverable Item Management Evaluator: Volume II: Program Listings and Narratives		5. TYPE OF REPORT & PERIOD COVERED INTERIM August 79-Jun 80
7. AUTHOR(s) W. Steven Denny		6. PERFORMING ORG. REPORT NUMBER TR-80-02-A
		8. CONTRACT OR GRANT NUMBER(s) C-0524 F33600-78-R-0314
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 3575 Charlene Drive Dayton, Ohio 45432		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 23041A5
11. CONTROLLING OFFICE NAME AND ADDRESS 2750th ABW/PMA BLDG 1, Area C Wright-Patterson AFB, Ohio		12. REPORT DATE May 1980
		13. NUMBER OF PAGES 298
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		16. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
17. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
18. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) A		
19. SUPPLEMENTARY NOTES		
20. KEY WORDS (Continue on reverse side if necessary and identify by block number) Recoverable item, multi-echelon, inventory/repair, simulation, METRIC, MOD-METRIC, AFLC		
21. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the Recoverable Item Management Evaluator (RIME), a FORTRAN simulation model for evaluating the relative cost-effectiveness of analytic optimization procedures proposed for use in Air Force Logistics Command recoverable item management systems. Major features of the model include (a) the use of actual Air Force demand histories to drive the model demand processes, (b) modeling of current Air Force statistical estimation procedures, and (c) modeling of the dynamic interactions among initial provisioning, replenishment and distribution policies. Volume II documents the programs for RIME.		

TABLE OF CONTENTS

SECTION

- I. Introduction
- II. Events Generator Programs
- III. Levels Computation System Programs
- IV. RIME Simulation Model Programs
- V. Job Control Language Files

Section I.

Introduction

This is the second of two volumes describing the features of the Recoverable Item Management Evaluator (RIME). RIME is a FORTRAN-based model of a multi-echelon inventory system consisting of a depot level repair facility, possibly several operating bases and an aircraft overhaul facility. The model was specifically designed to evaluate proposed inventory management policies for the control of Air Force Logistics Command recoverable item spares. Volume I of this report describes the philosophy, organization, and input/output features of RIME. This volume documents the statement listings of each RIME program, and provides narratives describing the function of each of these routines.

This Volume contains four major sections. Section I documents programs used in the RIME Exogenous Events Generator, while Section II documents programs used in the Levels Computation System. Section III documents programs in the RIME Simulation Model. Finally, Section IV documents Job Control Language files required to execute the RIME System.

Within each section, routines are presented in alphabetical order. Documentation of each routine consists of a program narrative and a listing of the FORTRAN code for that routine. Additional comments are included as a part of the FORTRAN code for the more complex routines. Definitions of major RIME variables are documented in Volume I, and familiarity with the contents of Volume I is assumed.

Section I
Events Generator Programs

List of Programs

Program

ASSIGN

BASEDA

EVTGNB

EVTGN2

LRUEV2

NOSET1

RANBAS

SCHIND

SRUEVT

SRUIND

READF2

SRUPRT

Subroutine: ASSIGN**Functions**

This routine determines the number of SRU failures to relate to the current LRU reparable generation event.

Calling Parameters:

- LBRGN = Total number of LRU reparable generations remaining to be considered.
- NOSRUF(K) = The total number of SRU K reparable generations remaining to be assigned to LRU reparable generations.
- NOINL(K) = The number of SRU K reparable generations to be assigned to the LRU reparable generation currently be considered.

Description:

Routine ASSIGN first computes the total number of SRU failure exposures SRUEX. This equals the total number of LRU reparable generations remaining (LBRGN) times the Quantity Per Application (IQPA(K)) for SRU K. The probability a specific unit of SRU K of the current LRU has failed is then $P = \text{NOSRUF}(K) / \text{SRUEX}$. Subroutine ASSIGN obtains a uniform random number R by calling subroutine RANDU, and compares this number to the probability P. If $R \leq P$, we

assume that the current SRU is related to the LRU failure and NOINL(K) is incremented by one. This Monte Carlo process is then repeated for each SRU component contained in the LRU. At the conclusion of the routine, NOINL(K) contains the number of reparable generations of SRU K to be associated with the current LRU failure.

SUBROUTINE ASSIGN(LRGN,NOSRUF,NOML)

5

COMMON/INT/INT(20)

COMMON/NSRU/NSRU

COMMON/IOPA/IOPA(1)

DIMENSION NOML(1),NOSRUF(1)

IB(INT(5).EQ.1) WRITE(6,17)

17 FORMAT(' ***** ENTER ASSIGN ***** ')

C

C LOOP THROUGH EACH SRU(K) IN THE LRU, WHERE K=1,2,...,NSRU

C

C DO 300 K=1,NSRU

C

C COMPUTE TOTAL SRU FAILURE EXPOSURES

C

C SRUX=FLOAT(LRGN*IOPA(K))

C

C INITIALIZE TO ZERO THE COUNTER OF THE NUMBER OF
C SRU(K) UNITS THAT HAVE FAILED IN THIS LRU.

C

C NOML(K)=0

C

C LOOK THROUGH EACH UNIT KK OF SRU(K) INSTALLED
C IN THE LRU, WHERE KK=1,2,...,IORT(K)

C

C KKK=IOPA(K)

C DO 300 KK=1,KKK

C

C COMPUTE THE FAILURE PROBABILITY P FOR THIS UNIT

C

C P=FLOAT(NOSRUF(K))/SRUX

C

C MONTE CARLO TO DETERMINE IF UNIT KK IS TO BE
C RELATED TO THE LRU FAILURE.

C

C R=RANDU(0.2)

C

C IF R LESS THAN OR EQUAL TO P, SET IFLAG=1 TO
C INDICATE THAT THIS SRU IS RELATED TO THE
C LRU FAILURE; OTHERWISE, SET IFLAG=0.

C

C IFLAG=0

C IF(R.LE.P) IFLAG=1

C

C IF(INT(5).EQ.1)WRITE(6,19)=ASSIGN, SRUX, KK, P, R, IFLAG

19 FORMAT(' ')

C IF(INT(5).EQ.1)WRITE(6,397)SRUX, KK, P, R, IFLAG

397 FORMAT('10.F5,0.I5,2F0.3,I5')

C

C IF IFLAG EQUALS 1, DECREMENT THE SRU'S FAILURES
C COUNTER BY 1 AND INCREASE COUNTER FOR THE
C TOTAL SRU(K) UNITS THAT HAVE FAILED IN THIS
C LRU, OTHERWISE GO TO NEXT STEP

C

ASSIGN

4 10-20-79 10.332

6

C

IF(IFLAG,EO,1) NOSRUP(K)=NOSRUP(K)-1

IF(IFLAG,EO,1) NOYML(K)=NOYML(K)+1

C

C

DECREMENT THE EXPOSURES COUNTER.

C

SRUX=SRUX-1

C

C

END OF KK LOOP FOR UNITS IN IOPA(K)

C

300 CONTINUE

400 CONTINUE

IF(IWT(5),EO,1) WRITE(6,10)

10 FORMAT("-----EXIT ASSIGN")

RETURN

END

Subroutine: BASEDA

7

Function:

This routine inputs order and ship times and flying program data by base.

Description:

This routine is used by both the Events Generator and the Level Computation System to input order and ship time and flying program data by base. If the write flag IWT(12) equals 1, the routine prints out all input information.

04 10-20-79 10,391 .S

8

*NRUN=:TIME/OBJ/BASEDA.O(RCD,NOGO)

*BASEDA,S

SUBROUTINE BASEDA

COMMON/NBASES/NBASES

COMMON/OSTDLT/OSTDLT(8)

COMMON/BFH/BFH(24,8)

COMMON/IWT/IWT(20)

CHARACTER IDENT*6

CALL PPARAM(1,131)

C

C-----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

THIS ROUTINE INPUTS ORDER AND SHIP TIMES IN
ADDITION TO FLYING PROGRAMS BY BASE WHERE

OSTDLT(K)=DEVIATIONS (IN DAYS) OF DEPOT
ORDER AND SHIP TIME AT BASE K
FROM THE AVERAGE ORDER AND SHIP
TIME FOR THE GROUP.

BFH(I,K)=FLYING HOURS AT BASE K DURING
QUARTER I.

INITIAL IDBUG.

IDBUG=IWT(12)

GTOTL=0.

IF(IDBUG.EQ.1)WRITE(6,13)*****ENTER BASEDA=--*

C-----*-----ADD A READ

C

READ(5,13,END=500) NBASES ,(OSTDLT(K),K=1,NBASES)
13 FORMAT(V)

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

REPORT SECTION.

IF(IDBUG.EQ.0) GO TO 20

WRITE(6,40)((X),I=1,NBASES)

40 FORMAT(//T10,"BASE NUMBER"/

&

/T15,8I10)

WRITE(6,50)(OSTDLT(K),K=1,NBASES)

50 FORMAT(/" OSTDLT ",T13,8(F8.0,2X))

20 CONTINUE

DO 100 I=1,24

READ(5,13,END=500) IDENT,(BFH(I,K),K=1,NBASES)

60 FORMAT(8X,A6,T13,8(F8.0,2X))

IF(IDBUG.NE.1) GO TO 100

COMPUTE TOTAL PROGRAM OVER ALL BASIS

BASEDA

04 10-28-79 10.391 .5

9

```
TOTAL=0.
DO 70 J=1,NBASES
TOTAL=TOTAL+BFH(I,J)
70 CONTINUE
GTOTL=GTOTL+TOTAL
WRITE(6,73)TOTAL
73 FORMAT(F95,"TOTAL =",F10.0)
WRITE(6,60) IDENT,(BFH(I,K),K=1,NBASES)
100 CONTINUE
500 CONTINUE
IF(IDBV9.GE.1)WRITE(6,503)GTOTL
503 FORMAT(//T90,"GRAND TOTAL=",F10.0//)
RETURN
END
```


Subroutine: EVTGN2, EVTGNB

Function:

This routine serves as the main program for the Events Generator. Program EVTGN2 generates exogenous events for LRU/SRU families that require no more than 2,000 events on the Future Events List (FEL) by any one time, while program EVTGNB is used for larger event families. At present, EVTGNB permits up to 3,000 events on the FEL at any given time.

Description:

For a detailed description of the Events Generator, see the Events Generator Chapter of Volume I.

```

1  *NRUNE;RTIME/OBJ/EVTGN2.O(BCD,NOGO)
2  *EVTGN2.S
3  C
4  C-----DEFINE GASP COMMONS-----
5  C
6      DIMENSION NSET(1)
7      COMMON QSET(500)
8      EQUIVALENCE(NSET(1),QSET(1))
9      COMMON /SCOM1/ ATRIB(30),JENNT,MFA,MFE(100),MLE(100),MSTOP,MCRDR,
10     &  MNAPO,MNAFT,MNATR,MNFIL,MNQ(100),MNTY,MNPT,PPARM(50,4),TMOV,TBEG,
11     &  TTCLR,TTFM,TTTB(30),TTSET
12      COMMON /SCOM6/ BENQ(100),IINN(100),KKRKK(100),MMAX3(100),
13     &  QOTIN(100),SSOBY(25,5),SETBY(25,6),VVNQ(100)
14  C
15  C-----
16      COMMON/IDBUG/IDBUG
17      COMMON/ITIME/ITIME
18      COMMON/NFGRP/NFGRP
19      COMMON/NGROUP/NGROUP
20      COMMON/NLGRP/NLGRP
21      COMMON/NBASES/NBASES
22      COMMON/NJOB/NJOB
23      COMMON/NSRU/NSRU
24      COMMON/NITEM/NITEM
25      COMMON/ROUT/ROUT
26      COMMON/IWT/IWT(20)
27  C
28      COMMON/OSTDLT/OSTDLT(8)
29      COMMON/BEH/BEH(24,8)
30      COMMON/CPROB/CPROB(8)
31  C
32      COMMON/MENTRY/MENTRY
33      COMMON/MFPMAX/MFPMAX
34      COMMON/MTIME/MTIME
35      COMMON/ILOCPE/ILOCPE(2000)
36      COMMON/JFSN/JFSN(2000)
37      COMMON/JPOINT/JPOINT(2000)
38      COMMON/JPRIOR/JPRIOR(2000)
39      COMMON/JQTY/JQTY(2000)
40      COMMON/JTIME/JTIME(2000)
41      COMMON/JTYPE/JTYPE(2000)
42  C
43      COMMON/IRSGN/IRSGN(16,40)
44      COMMON/IRTS/IRTS(16,40)
45      COMMON/IRCOND/IRCOND(16,40)
46      COMMON/INETS/INETS(16,40)
47      COMMON/IDCOND/IDCOND(16,40)
48      COMMON/IDREP/IDREP(16,40)
49      COMMON/IOVCND/IOVCND(16,40)
50      COMMON/IPPROG/IPPROG(16,40)
51      COMMON/IDRGN/IDRGN(16,40)
52  C

```

EVTGN2

53 COMMON/PSR/PSR(40)
 54 COMMON/UCOST/UCOST(40)
 55 COMMON/LTADM/LTADM(40)
 56 COMMON/LTPROD/LTPROD(40)
 57 COMMON/TBRT/TBRT(40)
 58 COMMON/IBDTT/IBDTT(40)
 59 COMMON/IDRT/IDRT(40)
 60 COMMON/IOST/IOST(40)
 61 COMMON/OTMDDR/OTMDDR(40)
 62 COMMON/RTSF/RTSF(40)
 63 COMMON/PDRF/PDRF(40)
 64 COMMON/EOHF/EOHF(40)
 65 COMMON/BMSTRF/BMSTRF(40)
 66 COMMON/DPDDR/DPDDR(40)
 67 COMMON/PJRCND/PJRCND(40)
 68 COMMON/PNJRC/PNJRC(40)
 69 COMMON/OTMBRR/OTMBRR(40)
 70 COMMON/BCONF/BCONF(40)
 71 COMMON/EJCNF/EJCNF(40)
 72 COMMON/ENCNF/ENCNF(40)
 73 COMMON/DOCNF/DOCNF(40)
 74 COMMON/RMSCNF/RMSCNF(40)
 75 COMMON/RMHRF/RMHRF(40)
 76 COMMON/RCOST/RCOST(40)
 77 COMMON/IRPD/IRPD(40)
 78 COMMON/IRINTR/IRINTR(40)
 79 COMMON/IOPA/IOPA(40)

C

80 COMMON/MOIAL/MOIAL(40)
 81 COMMON/NBRTS/NBRTS(40)
 82 COMMON/MNRTS/MNRTS(40)
 83 COMMON/NBCOND/NBCOND(40)
 84 COMMON/MOVCNT/MOVCNT(40)
 85 COMMON/MODRPT/MODRPT(40)
 86 COMMON/MOSRUF/MOSRUF(40)
 87 COMMON/NDRGNT/NDRGNT(40)
 88 COMMON/NDCOND/NDCOND(40)
 89 COMMON/NDCONT/NDCONT(40)
 90 COMMON/MOVCNT/MOVCNT(40)
 91 COMMON/NDEXFO/NDEXFO(40)

C

92 COMMON/LRTS/LRTS
 93 COMMON/LNRTS/LNRTS
 94 COMMON/LBRGN/LBRGN
 95 COMMON/LBCOND/LBCOND
 96 COMMON/LDRGN/LDRGN
 97 COMMON/LDRGNT/LDRGNT
 98 COMMON/LDREPT/LDREPT
 99 COMMON/LNRTST/LNRTST
 100 COMMON/LOVCNT/LOVCNT
 101 COMMON/LOYGNT/LOYGNT
 102 COMMON/LDCONT/LDCONT

```

5      COMMON/LDCOND/LDCOND
6      C
7      COMMON/INQTR/INQTR
8      COMMON/ITDAY/ITDAY
9      COMMON/ITMTH/ITMTH
10     COMMON/ITQTR/ITQTR
11     COMMON/ITYEAR/ITYEAR
12     COMMON/ITINV/ITINV
13     C
14     NAMELIST/DATA1/LNRTST,LDRGNT,LDCONT,LOVGNT,LOYCNT,LDREPT,NSBU,NITEM
15     C
16     CHARACTER FSN*15
17     C
18     C
19     C      READ SIMULATION PARAMETERS
20     WRITE(6,13) " NFGRP, NLGRP, NBASES, INQTR, NREPL"
21     READ(5,13) NFGRP, NLGRP, NBASES, INQTR, NREPL
22     WRITE(6,23) NFGRP, NLGRP, NBASES, INQTR, NREPL
23     FORMAT(10I7)
24     C
25     C
26     READ(5,13) IWT
27     WRITE(6,34) IWT
28     3 FORMAT(// "WRITE FLAGS=" ,4(5X,5I2)//)
29     C
30     IPRNT=IWT(8)
31     IPRN2=IWT(9)
32     IDBUG=IWT(10)
33     IOUT=IWT(11)
34     C
35     C      INPUT BASE DATA
36     C
37     CALL BAMEDA
38     C
39     C
40     C      WRITE EXOGENOUS FILE HEADER RECORD
41     C
42     C      RESET NBASES TO CORRESPOND TO INPUT PARAMETER
43     C
44     C
45     NBASES=KBASES
46     IF(IOUT.NE.1) WRITE(8) NFGRP, NLGRP, NBASES, INQTR, NREPL
47     C
48     C
49     R=RANDU(0,1)
50     ISEQ=
51     KNTOUT=0
52     C
53     C
54     C      SET F.E.L. LIMITS
55     C
56     MAXFEL=0

```

T 01 11-02-79 15.594 .5

14

```
57      NPEMAX=2000
58      C
59      C
60      C      SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY
61      C      7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR
62      C
63      ITDAY=100
64      ITWEEK=7*ITDAY
65      ITMNTN=4*ITWEEK
66      ITQTR=3*ITMNTN
67      ITYEAR=4*ITQTR
68      C
69      C      SKIP THE FIRST (NFGRP-1) LRU GROUPS ON FILE 07
70      CALL READF1
71      NSKIP=NFGRP-1
72      IF(NSKIP.LE.0) GO TO 60
73      DO 50 NG=1,NSKIP
74      NGROUP=NG
75      CALL READF1(IENDF)
76      IF(IENDF.EQ.1) GO TO 980
77      50 CONTINUE
78      60 CONTINUE
79      C
80      C
81      C      BEGIN GROUP LOOP
82      C
83      DO 10 NG=NFGRP,NLGRP
84      NGROUP=NG
85      WRITE(6,13)"BEGIN LRU GROUP=",NG
86      13 FORMAT(V)
87      C
88      C      SET LRU AND SRU HEADER DATA
89      C
90      CALL READF1(IENDF)
91      IF(IENDF.EQ.1) GO TO 980
92      C      BEGIN REPLICATION LOOP
93      C
94      DO 99 KREPL=1,NREPL
95      IF(IPRN2.EQ.1)WRITE(6,13)"BEGIN REPLICATION=",KREPL
96      C
97      C
98      C      RESET JOBS COUNTER
99      C
100     NJOB=1 00
101     C
102     C      INITIALIZE THE FUTURE EVENTS LIST
103     C
104     CALL INFE1
105     C
106     C      INITIALIZE DEPOT
107     C      EVENT COUNTERS
108     C
```

```

09      NNRUNITER=1
10      C
11      CALL NQZB1
12      IF(IPRNT.EQ.1) WRITE(6,DATA1)
13      C
14      C
15      C-----
16      C      BEGIN QUARTER LOOP
17      C-----
18      C
19      DO 800 NQZB=1,INQTR
20      ITINV=IQTR
21      C
22      C      INITIALIZE BASE PROBABILITY ARRAY CPROB.
23      C
24      CALL RANBS1(ITINV)
25      C
26      C
27      C      INITIALIZE BASE EVENT COUNTERS
28      C
29      CALL NQZB2(ITINV)
30      C      LOOP THROUGH EACH LRU FAILURE(L), WHERE L = 1,2,...,LDRGN
31      C
32      IF(IPRNT.EQ.1) CALL SRUPRT
33      C
34      C
35      C-----BEGIN LRU BASE CONDEMNATION AND RTS GENERATION
36      C
37      IF(LDRGN.GT.0)CALL LRUEVT(LDRGN.LRTS,NJOB,1)
38      C
39      C      GENERATE LRU NRTS EVENTS IF ANY
40      C
41      IF(LNRTS.GT.0)CALL LRUEVT(LNRTS.NLRUN,NJOB,2)
42      C
43      C      GENERATE LRU DEPOT REP GEN EVENTS, IF ANY
44      C
45      NTOOV=LDRGN-LDCOND
46      IF(LDRGN.GT.0)CALL LRUEVT(LDRGN.NTOOV,NJOB,3)
47      C
48      C-----END OF LRU EVENT GENERATION-----
49      C
50      510 CONTINUE
51      IF(IPRNT.NE.1) GO TO 580
52      WRITE(6,DATA1)
53      CALL SRUPRT
54      580 CONTINUE
55      C
56      C      GENERATE INDEPENDENT SPU EVENTS, IF ANY
57      C
58      CALL SRUIND
59      C
60      C

```

```

61      C      REMOVE ALL EVENTS FROM THE FEL. AND WRITE
62      C      THEM TO THE EXOGENOUS EVENT FILE. AT END,
63      C      GO TO 700
64      60 CONTINUE
65      C
66      C      RECORD MAX ENTRIES TO THE F.E.L.
67      C
68      MAXFEL=MAX(MAXFEL,NENTRY)
69      C
70      C      REMOVE EVENTS FROM THE F.E.L.
71      C
72      JOEND=IOTR*ITOTR
73      IF(NENTRY.LE.0) GO TO 700
74      IF(NTIME.GT.JOEND) GO TO 700
75      CALL REMOVE(KTIME,KTYPE,KFSN,KQTY,KPRI)
76      IF(IPRN2.EQ.1)WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
77      617 FORMAT("REMOVE*****",818)
78      IF(IOUT.EQ.1)WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
79      IF(IOUT.EQ.1)KNTOUT=KNTOUT+1
80      GO TO 600
81      700 CONTINUE
82      C
83      C      END OF QUARTER LOOP
84      C
85      IF(IPRNT.EQ.1) WRITE(6,703)(IOTR,I=1,8)
86      703 FORMAT( 8("----END QTR",I3)/2425("----")/)
87      C
88      80 CONTINUE
89      C
90      C      SCHEDULE TYPE 10 EVENT...."END OF SIMULATED PERIOD"
91      C
92      KTIME=99999999
93      KTYPE=10
94      KFSN=0
95      KQTY=
96      KPRI=
97      IF(IPRN2.EQ.1)WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
98      IF(IOUT.EQ.1)WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
99      IF(IOUT.EQ.1)KNTOUT=KNTOUT+1
100     C
101     C      END OF REPLICATION LOOP
102     990 CONTINUE
103     C
104     C      END OF GROUP LOOP
105     C
106     1000 CONTINUE
107     C
108     C      END OF DATA SET GENERATION
109     C
110     REWIND 08
111     WRITE(6,13)"END OF EVENT SET GENERATION"
112     WRITE(6,13)"MAX ENTRIES TO THE F.E.L. =",MAXFEL

```

13 WRITE(6,13)"NO. OF EVENTS WRITTEN TO EXOG FILE =".

14 & KNTOUT

15 C

16 STOP

17 99 CONTINUE

18 C

19 REWIND 08

20 WRITE(6,13)"READ END-OF-FILE ON 07. STOP RUN."

21 WRITE(6,13)"LAST LRU READ WAS=I,FSN(1)"

22 STOP

23 END

*W 7 MEMORY EXPANDED. USE \$LIMITS OR CORE= OPTION FOR NEXT RUN

01 11-07-79 14.521 .S

```

1  *#RUN=;TIME/OBJ/EVTGNE.(BCD,NOGO)
2  *EVTGN2.S
3  C
4  C-----DEFINE GASP COMMONS-----
5  C
6      DIMENSION NSET(1)
7      COMMON QSET(500)
8      EQUIVALENCE(NSET(1),QSET(1))
9      COMMON /GCOM1/ ATZB(30),JPVNT,MFA,MFE(100),MLE(100),MSTOP,MCRDR,
0  A  MNAPO,MNAPT,MNATR,MNFIL,MNQC(100),MNTY,MNPT,MNPM(50,4),TNOW,TTWES,
1  A  TTCLR,TTFIN,TTIB(30),TTSET
2      COMMON /GCOM6/ EENQ(100),IIN(100),KKRANK(100),MNAHQ(100),
3  A  QLTIN(100),SSOBV(25,5),SSPV(25,6),VVNC(400)
4  C
5  C-----
6      COMMON/IDBUG/IDBUG
7      COMMON/ITIME/ITIME
8      COMMON/MFGRP/MFGRP
9      COMMON/MGROUP/MGROUP
0      COMMON/MLGRP/MLGRP
1      COMMON/MBASES/MBASES
2      COMMON/MJOB/MJOB
3      COMMON/MSRU/MSRU
4      COMMON/MITEM/MITEM
5      COMMON/ICUT/ICUT
6      COMMON/INT/INT(20)
7  C
8      COMMON/OSTELT/OSTELE(8)
9      COMMON/BPH/BPH(24,8)
0      COMMON/CPROB/CPROB(8)
1  C
2      COMMON/MENTRY/MENTRY
3      COMMON/MPEMAX/MPEMAX
4      COMMON/MTIME/MTIME
5      COMMON/TIOCFE/TIOCFE(3000)
6      COMMON/JTSH/JTSH(3000)
7      COMMON/JPOINT/JPOINT(3000)
8      COMMON/JPRIOR/JPRIOR(3000)
9      COMMON/JQTY/JQTY(3000)
0      COMMON/JTIME/JTIME(3000)
1      COMMON/JTYPE/JTYPE(3000)
2  C
3      COMMON/IBRGN/IBRGN(16,40)
4      COMMON/IRTS/IRTS(16,80)
5      COMMON/IBCOND/IBCOND(16,40)
6      COMMON/INRTS/INRTS(16,40)
7      COMMON/IDCOND/IDCOND(16,40)
8      COMMON/IDREP/IDREP(16,40)
9      COMMON/IOVCHD/IOVCHD(16,40)
0      COMMON/IPPROG/IPPROG(16,40)
1      COMMON/IDRGN/IDRGN(16,40)
2  C

```

EVTGNB

01 11-07-79 14,521 .5

3 COMMON/PSN/PSN(40)
 4 COMMON/UCOST/UCOST(40)
 5 COMMON/LTADM/LTADM(40)
 6 COMMON/LTPROD/LTPROD(40)
 7 COMMON/IBRT/IBRT(40)
 8 COMMON/IBDIT/IBDIT(40)
 9 COMMON/IDRT/IDRT(40)
 10 COMMON/IOST/IOST(40)
 11 COMMON/OIMDB/OIMDB(40)
 12 COMMON/RTSP/RTSP(40)
 13 COMMON/PDMF/PDMF(40)
 14 COMMON/CONP/CONP(40)
 15 COMMON/RMSTP/RMSTP(40)
 16 COMMON/PPDDR/PPDDR(40)
 17 COMMON/PJRCND/PJRCND(40)
 18 COMMON/RMJRC/RMJRC(40)
 19 COMMON/OIMBER/OIMBER(40)
 20 COMMON/BCONP/BCONP(40)
 21 COMMON/EJCNE/EJCNE(40)
 22 COMMON/ENCNE/ENCNE(40)
 23 COMMON/DOHCNE/DOHCNE(40)
 24 COMMON/RMSCNE/RMSCNE(40)
 25 COMMON/RMNEP/RMNEP(40)
 26 COMMON/ECOST/ECOST(40)
 27 COMMON/IBPD/IBPD(40)
 28 COMMON/IRINT/IRINT(40)
 29 COMMON/IOPA/IOPA(40)

C

11 COMMON/NOINL/NOINL(40)
 12 COMMON/MBRTS/MBRTS(40)
 13 COMMON/MBRTS/MBRTS(40)
 14 COMMON/MBCOND/MBCOND(40)
 15 COMMON/NOVCNT/NOVCNT(40)
 16 COMMON/MODRPT/MODRPT(40)
 17 COMMON/NOSEUP/NOSEUP(40)
 18 COMMON/MDRGT/MDRGT(40)
 19 COMMON/MDCOND/MDCOND(40)
 20 COMMON/MDCONT/MDCONT(40)
 21 COMMON/NOVENT/NOVENT(40)
 22 COMMON/INDEXO/INDEXO(40)

C

14 COMMON/LRTS/LRTS
 15 COMMON/LNRTS/LNRTS
 16 COMMON/LBRCN/LBRCN
 17 COMMON/LBCOND/LBCOND
 18 COMMON/LDRGN/LDRGN
 19 COMMON/LDRGT/LDRGT
 20 COMMON/LDRPT/LDRPT
 21 COMMON/LNRTST/LNRTST
 22 COMMON/LOVCNT/LOVCNT
 23 COMMON/LOVENT/LOVENT
 24 COMMON/LDCONT/LDCONT

01 11-07-79 14.521 .5

```

5      COMMON/LDCOND/LDCOND
6      C
7      COMMON/INOTR/INOTR
8      COMMON/ITDAY/ITDAY
9      COMMON/ITMNTH/ITMNTH
10     COMMON/ITOTR/ITOTR
11     COMMON/ITYEAR/ITYEAR
12     COMMON/ITINV/ITINV
13     C
14     NAMELIST/DATA1/LNVTST,LPRGNT,LDCONT,LOVGNT,LCVCNT,LDRFPT,NSRU,NITPM
15     C
16     CHARACTER PSN*15
17     C
18     C
19     C      READ SIMULATION PARAMETERS
20     WRITE(0,13) NFGRP, NLGRP, KBASES, INOTR, NREPL
21     READ(5,13) NFGRP, NLGRP, KBASES, INOTR, NREPL
22     WRITE(0,23) NFGRP, NLGRP, KBASES, INOTR, NREPL
23     FORMAT(10Y7)
24     C
25     C
26     READ(5,13) IWT
27     WRITE(0,3) IWT
28     3 FORMAT(/"WRITE FLATS=",4(5X,5I2)/)
29     C
30     IPENT=INT(8)
31     IPEN2=INT(9)
32     IBUG=INT(10)
33     IOUT=INT(11)
34     C
35     C      INPUT BASE DATA
36     C
37     CALL BASEDA
38     C
39     C      WRITE EXOGENOUS FILE HEADPR RECORD
40     C
41     C      RESET KBASES TO CORRESPOND TO INPUT PARAMETER
42     C
43     C
44     C      KBASES=KBASES
45     IF(IOUT, EQ, 1) WRITE(0) NFGRP, NLGRP, KBASES, INOTR, NREPL
46     C
47     C
48     C
49     R=RANDU(-,1)
50     ISEQ=0
51     KNTOUT=0
52     C
53     C
54     C      SET F.E.L. LIMITS
55     C
56     MAXFEL=0

```

01 11-07-79 14.521 .S

17 NFEMAX=3000

18 C

19 C

20 C SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY

21 C 7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR

22 C

23 ITDAY=100

24 ITWEEK=7*ITDAY

25 ITMONTH=4*ITWEEK

26 ITQTR=3*ITMONTH

27 ITYEAR=4*ITQTR

28 C

29 C SKIP THE FIRST (N*GRP-1) LRU GROUPS ON FILE 07

30 CALL READFL

31 NSKIP=NSGRP-1

32 IF(NSKIP.LE.0) GO TO 60

33 DO 50 NG=1,NSKIP

34 NGROUP=NG

35 CALL READFL(IENDF)

36 IF(IENDF.EQ.1) GO TO 980

37 50 CONTINUE

38 60 CONTINUE

39 C

40 C

41 C BEGIN GROUP LOOP

42 C

43 DO 1000 NG=NSGRP,NLGRP

44 NGROUP=NG

45 WRITE(6,13)*BEGIN LRU GROUP="NG

46 13 FORMAT(V)

47 C

48 C SET LRU AND SRU HEADER DATA

49 C

50 CALL READFL(IENDF)

51 IF(IENDF.EQ.1) GO TO 980

52 C BEGIN REPLICATION LOOP

53 C

54 DO 990 KREPL=1,KREPL

55 IF(IPW2.EQ.1)WRITE(6,13)*BEGIN REPLICATIONS="KREPL

56 C

57 C

58 C RESET JOBS COUNTER

59 C

60 NJOB=1000

61 C

62 C INITIALIZE THE FUTURE EVENTS LIST

63 C

64 CALL INFEL

65 C

66 C INITIALIZE D*POT

67 C EVENT COUNTERS

68 C

```

09      NSNU=HITEM-1
10      C
11      CALL NOSET1
12      IF(IPRNT.EQ.1) WRITE(6,DATA1)
13      C
14      C
15      C-----
16      C      BEGIN QUARTER LOOP
17      C-----
18      C
19      DO 200 IQTR=1,IQTR
20      ITINV=IQTR
21      C
22      C      INITIALIZE BASE PROBABILITY ARRAY CPROB,
23      C
24      CALL RANBS1(ITINV)
25      C
26      C
27      C      INITIALIZE BASE EVENT COUNTERS
28      C
29      CALL NOSET2(ITINV)
30      C      LOOP THROUGH EACH LRU FAILURE(L), WHERE L = 1,2,...,LBRGN
31      C
32      IF(IPRNT.EQ.1) CALL SHUPRT
33      C
34      C
35      C-----BEGIN LRU BASE CONDEMNATION AND HIS GENERATION
36      C
37      IF(LBRGN.GT.0)CALL LBUEVT(LBRGN,LRTS,NJOB,4)
38      C
39      C      GENERATE LRU WRTS EVENTS, IF ANY
40      C
41      IF(LNRTS.GT.0)CALL LBUEVT(LNRTS,NLRUN,NJOB,2)
42      C
43      C      GENERATE LRU DEPOT REP GEN EVENTS, IF ANY
44      C
45      NTOOV=LBRGN-LDCOND
46      IF(LDEGN.GT.0)CALL LBUEVT(LBRGN,NTOOV,NJOB,3)
47      C
48      C=====END OF LRU EVENT GENERATION=====
49      C
50      510 CONTINUE
51      IF(IPRNT.NE.1) GO TO 580
52      WRITE(6,DATA1)
53      CALL SHUPRT
54      580 CONTINUE
55      C
56      C      GENERATE INDEPENDENT SRU EVENTS, IF ANY
57      C
58      CALL SHUIND
59      C
60      C

```

C 01 11607-79 14.521 JS

```

51      C      REMOVE ALL EVENTS FROM THE FEL, AND WRITE
52      C      THEM TO THE EXOGENOUS EVENT FILE, AT END,
53      C      GO TO 700
54      600 CONTINUE
55      C
56      C      RECORD MAX ENTRIES TO THE F.E.L.
57      C
58      MAXFEL=MAX(MAXFEL,NENTRY)
59      C
60      C      REMOVE EVENTS FROM THE F.E.L.
61      C
62      JOEND=IQTR*ITQTR
63      IF(NENTRY.LE.0) GO TO 700
64      IF(NTIME.GT.JOEND) GO TO 700
65      CALL REMOVE(KTIME,KTYPE,KFSN,KQTY,KPRI)
66      IF(IPRNT.EQ.1)WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
67      517 FORMAT('REMOVE*****',818)
68      IF(IOUT.EQ.1)WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
69      IF(IOUT.EQ.1)KNTOUT=KNTOUT+1
70      GO TO 600
71      700 CONTINUE
72      C
73      C      END OF QUARTER LOOP
74      C
75      IF(IPRNT.EQ.1) WRITE(6,703)(IQTR,I=1,8)
76      703 FORMAT( 8('-----END QTR',I3)/2(25('-----')/))
77      C
78      800 CONTINUE
79      C
80      C      SCHEDULE TYPE 10 EVENT,..."END OF SIMULATED PERIOD"
81      C
82      KTIME=999999999
83      KTYPE=10
84      KFSN=0
85      KQTY=0
86      KPRI=0
87      IF(IPRNT.EQ.1)WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
88      IF(IOUT.EQ.1)WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
89      IF(IOUT.EQ.1)KNTOUT=KNTOUT+1
90      C
91      C      END OF REPLICATION LOOP
92      990 CONTINUE
93      C
94      C      END OF GROUP LOOP
95      C
96      1000 CONTINUE
97      C
98      C      END OF DATA SET GENERATION
99      C
100     REMIND 08
101     WRITE(6,13)"END OF EVENT SET GENERATION"
102     WRITE(6,13)"MAX ENTRIES TO THE F.E.L. =",MAXFEL

```

T 01 11-07-79 14.521 .S

```
13      WRITE(0,13)"NO. OF EVENTS WRITTEN TO EXOS FILE =",  
14      LAYOUT  
15      C  
16      STOP  
17      980 CONTINUE  
18      C  
19      REWIND 04  
20      WRITE(0,13)"HEAD ENF-OF-FILE ON 07. STOP RUN."  
21      WRITE(0,13)"LAST 'R' HEAD WAS=",ENH(1)  
22      STOP  
23      END  
*6      7 MEMORY EXPANDED. USE SLIMITS OR CORP= OPTION FOR NEXT RUN
```

Subroutine: LRUEVT**Functions**

This routine controls the generation of all LRU exogenous events for the current quarter.

Calling Parameters:

- NOLEFT** = Total number of LRU reparable generations to be simulated for the current quarter.
- LRTS** = The number of these reparable generations that are to be repaired at the current location.
- NJOB** = The last job number assigned to a reparable generation.
- MODEF** = Mode of Failure Flag, where (1) indicates base condemnation or RTS events, (2) indicates a base NRTS event, and (3) indicates a depot rep gen event.

Description:

For a discussion of this routine, see the Events Generator Chapter of Volume I.


```

*SRUN=TIME/0BJ/LRUEV2.0(BCD,NOGO)
*LRUEV2,5 GENERATE LRU AND SRU EVENTS
*****GENERATE EXOGENOUS EVENTS FOR THE CURRENT QUARTER
SUBROUTINE LRUEVT(MOLEFT,LRTS,NJOB,MODEP)
C GENERATE LRU EVENTS, WHERE MODEP=
C 1 = BASE CONDEMNATION AND RTS EVENTS
C 2 = BASE RTS EVENTS
C 3 = DEPOT REP GEN EVENTS
C
COMMON/IWT/IWT(20)
COMMON/MSRU/MSRU
COMMON/IQPA/IQPA(1)
COMMON/NOINL/NOINL(1)
COMMON/MBRTS/MBRTS(1)
COMMON/MBRTS/MBRTS(1)
COMMON/MBCOND/MBCOND(1)
COMMON/NOVCNT/NOVCNT(1)
COMMON/NOVRUP/NOVRUP(1)
COMMON/IDBT/IDBT(1)
COMMON/IBDTT/IBDTT(1)
COMMON/IBRT/IBRT(1)
COMMON/LDCOND/LDCOND
COMMON/LPCONT/LPCONT
COMMON/LDRGN/LDRGN
COMMON/LDRGNT/LDRGNT
COMMON/NLRUP/NLRUP
COMMON/LOVCNT/LOVCNT
COMMON/LOVGNT/LOVGNT
COMMON/LDREPT/LDREPT
COMMON/LFRTST/LFRTST
C
C
COMMON/ITOTE/ITOTE
COMMON/ITDAY/ITDAY
COMMON/NBASES/NBASES
COMMON/ITINV/ITINV
COMMON/ITIME/ITIME
IF(IWT(4).EQ.1)WRITE(6,13)*****ENTER LRUEVT--MOLEFT=",
MOLEFT," LRTS=",LRTS," NJOB=",NJOB," MODEP=",MODEP,
" LOVCNT=",LOVCNT," LOVGNT=",LOVGNT," LDREPT",
LDREPT
13 FORMAT(V)
C
C IQTR=ITINV
C
C START EVENTS GENERATION LOOP
NUMB=MOLEFT
DO 9020 LLL=1,NUMB
C
C INCREMENT JOB COUNTER
C
C NJOB=NJOB+1

```

LRUEVT

```

3      C
4      C
5      C
6      IF(IWT(4),EQ,1)WRITE(6,323)NOLEPT,LLL,NJCB,MODEP
7      323 FORMAT(" NOLEPT=",I5," LRU=",I5," NJOB=",I5,
8      " MODEP=",I5)
9      S CONTINUE
10     C
11     C
12     C      MONTE CARLO TO DETERMINE THE BASE NUMBER AND LRU STOCK
13     C      KEEPING UNIT ASSOCIATED WITH THIS FAILURE.
14     C
15     C      CALL RANBAS(NB)
16     C
17     IF(MODEP,EQ,1)NSKLRLU=NB+1
18     IF(MODEP,EQ,3) NSKLRLU=NBASES+2
19     IF(MODEP,EQ,2)NSKLRLU=NB+1
20     IF(IWT(4),EQ,1)WRITE(6,13) " NR=",NB," NSKLRLU=",NSKLRLU
21     C
22     C      COMPUTE FAILURE TIME FOR THIS LRU, ASSUMING
23     C      FAILURE EVENTS ARE UNIFORMLY DISTRIBUTED THROUGHOUT
24     C      A QUARTER.
25     C
26     ITT=RANDU(.2)*ITQTR
27     ITIME=(IQTR-1)*ITQTR + ITT
28     C
29     C      SCHEDULE THE FAILURE EVENT FOR THIS LRU
30     C
31     JTIME=ITIME
32     CALL ENTER(JTIME,14,NSKLRLU,1,NJOB)
33     C
34     C
35     C      SCHEDULE A REQUISITION TO REPLACE THE FAILED LRU
36     C
37     JTIME=ITIME + 10
38     CALL ENTER(JTIME,1,NSKLRLU,1,1)
39     C
40     C      BRANCH BASED ON FAILURE MODE(MODEP).
41     C
42     GO TO (100,700,600).MODEP
43     C
44     C      RTS EVENT
45     C
46     C
47     100 CONTINUE
48     CALL ASSIGN(NOLEPT,NOSEUP,NOINL)
49     IF(IWT(4),EQ,1)WRITE(6,327) (NOINL(KKK),KKK=1,NSRU)
50     327 FORMAT(250,"NOINL(K)=",40I2)
51     C      AT THIS POINT, THE VARIABLE NOINL(K) EQUALS THE NUMBER
52     C      OF BASE REPARABLE GENERATIONS THAT ARE TO BE ASSOCIATED
53     C      WITH THE CURRENT LRU REPARABLE GENERATION. THE FOLLOWING
54     C      LOGIC FURTHER DEFINES THE FAILURE TYPES, AND SCHEDULES

```

```

1      C      ALL RELATED REQUISITION, REPAIR, AND CONDEMNATION ACTIONS:
2      C
3      C
4      C      FAILURE REMAINS AT THIS LOCATION
5      C
6      C
7      C      MONTE CARLO TO DETERMINE IF THE LRU IS REPARABLE:
8      C      IF NOT, GO TO 300:
9      C
10     C      LRUREP=1
11     C      R=RANDU(.2)
12     C      PRTS=FLOAT(LRTS)/FLOAT(NOLEFT)
13     C      IF(R.GT,PRTS)LRUREP=0
14     C
15     C      IF(INT(4),EQ,1)WRITE(6,13)'LRTS=',LRTS,' NOLEFT=',NOLEFT,
16     C      ' PRTS=',PRTS,' LRUREP=',LRUREP
17     C      IF(INT(4),EQ,1) WRITE(6,14)
18     C      14 FORMAT(' .....NOTE LRUREP=0 MEANS LRU IS CONDEMNED.')
19     C
20     C      IF(R.LE,PRTS)LRTS=LRTS-1
21     C
22     C      COMPUTE THE TOTAL PARTS (NNEED) NEEDED TO REPAIR THIS LRU.
23     C
24     C      NNEED=0
25     C      DO 110 K=1,NSRU
26     C      NNEED=NNEED+NOINL(K)
27     C      110 CONTINUE
28     C
29     C      IF LRU IS TO BE CONDEMNED, GO TO 300.
30     C      IF(LRUREP.LE,0) GO TO 300
31     C
32     C      IF NO PARTS ARE NEEDED (IE. IF NNEED=0), SCHEDULE AN LRU
33     C      REPAIR COMPLETION EVENT FOR STOCKKEEPING NUMBER NSKL RU
34     C
35     C      IF(NNEED.GE,1) GO TO 300
36     C      JTIME=ITIME+IBRT(NSKL RU)*ITDAY
37     C      CALL ENTER(JTIME,18,NSKL RU,1,NJOB)
38     C
39     C      COMPLETE LRU EVENT SCHEDULING.
40     C
41     C      GO TO 4000
42     C
43     C      THERE HAS BEEN AN SRU FAILURE. THE FOLLOWING LOGIC
44     C      SCHEDULES ALL RELATED RTS, NRTS, CONDEMNATION,
45     C      AND REQUISITION AND REPAIR COMPLETION EVENTS.
46     C
47     C
48     C      200 CONTINUE
49     C
50     C      SCHEDULE "A BEGIN WAIT" EVENT FOR THE LRU
51     C
52     C

```

```

7      JTIME=JTIME + 5
8      CALL ENTER(JTIME,16,N$KL$RU,N$NEED,N$JOB)
9      C
0      C
1      C      SCHEDULE ALL RELATED SRU EVENTS
2      C
3      CALL S$UEVT(L$UREP,N$KL$RU)
4      GO TO 9000
5      C
6      C      LRU IS TO BE CONDEMNED
7      C
8      300 CONTINUE
9      C
0      C      SCHEDULE LRU CONDEMNATION EVENT(EVENT CODE 15)
1      C
2      JTIME=JTIME+10
3      CALL ENTER(JTIME,15,N$KL$RU,1,N$JOB)
4      C
5      C      SCHEDULE ALL RELATED SRU EVENTS
6      C
7      CALL S$UEVT(L$UREP,N$KL$RU)
8      GO TO 9000
9      C
0      C
1      C*****LRU DEPOT REP GEN
2      C
3      600 CONTINUE
4      C
5      C      MONTE CARLO TO DETERMINE IF REP GEN IS
6      C      A DEPOT CONDEMNATION
7      C      IF NOT, GO TO 720 AND WRTE THIS LRU TO THE DEPOT
8      C
9      C
0      DCONB=FLOAT(LDCOND)/FLOAT(NOLEFT)
1      R=RANPU(.2)
2      C
3      IF(INT(4),EQ,1)WRITE(6,13)*****DEPOT REP GEN*
4      "      LDRGNT=",LDRGNT,"      LDRGN=",LDRGN,
5      "      LDCOND=",LDCOND,"      DCONB=",DCONB,
6      "      R=",R
7      C
8      C      UPDATE TOTAL COUNTS FOR DEPOT REP GENs
9      C
0      LDRGNT=LDRGNT-1
1      C
2      IF(R.GE,DCONB) GO TO 710
3      C
4      C      SCHEDULE A DEPOT CONDEMNATION EVENT
5      C
6      LDGOND=LDCOND-1
7      LDGONT=LDGONT-1
8      JTIME=JTIME+20

```

```

8      CALL ENTER(JTIME,15,NSKLBU,1,NJOB)
9      C
10     GO TO 9000
11     C
12     C
13     C
14     C=====LRU NRTS EVENTS
15     C
16     700 CONTINUE
17     C
18     C      SCHEDULE NRTS EVENT
19     C
20     LVRTST=LVRTST-1
21     710 CONTINUE
22     JTIME=JTIME+30
23     CALL ENTER(JTIME,19,NSKLBU,1,NJOB)
24     C
25     C      MONTE CARLO TO DETERMINE IF LRU GENERATION IS
26     C      OVERHAUL CONDEMNED
27     C
28     720 CONTINUE
29     DCONR=FLOAT(LOVCNT)/FLOAT(LDREPT+LOVCNT)
30     R=RANDU(.2)
31     C
32     IF(INT(4).EQ.1)WRITE(6,13)'<<<<<NRTS EVENT',
33     &      " LOVCNT=",LOVCNT," LDREPT=",LDREPT,
34     &      " DCONR=",DCONR," R=",R
35     C
36     IF(DCONR.LE.R) GO TO 750
37     C
38     C      SCHEDULE DEPOT OVERHAUL CONDEMNATION EVENT.
39     C
40     LOVCNT=LOVCNT-1
41     LOVCNT=LOVCNT-1
42     JTIME=JTIME+IBDTT(1)+ITDAY
43     CALL ENTER(JTIME,15,1,1,NJOB)
44     GO TO 9000
45     C
46     C      SCHEDULE DEPOT REPAIR COMPLETION
47     C
48     750 CONTINUE
49     LDREPT=LDREPT-1
50     LOVCNT=LOVCNT-1
51     JTIME=JTIME+(IBDTT(1)+IBRT(1))+ITDAY
52     CALL ENTER(JTIME,18,1,1,NJOB)
53     GO TO 9000
54     C
55     C
56     C=====END OF EVENTS LOOP
57     9000 CONTINUE
58     C
59     C      DECREASE NUMBER OF LRUS REMAINING.
60     C

```

Subroutine: NOSET1, NOSET2

Functions

This routine initializes counter arrays that determine the number of LRU and SRU events to be generated. Entry point NOSET1 initializes depot event counters, while NOSET2(KQ) initializes counters for quarter KQ.

Descriptions

In this routine, the variable N denotes an item number. The LRU is denoted by N=1, while item numbers of N=2, N=3, ... correspond to the first SRU, the second SRU, and so on. For definitions of variables used in this routine, see the Events Generator Chapter of Volume I.

Entry point NOSET1 initializes depot events counters. The routine first computes the total number of LRU depot events associated with the variables LDCONT, LOVCNT, LNRTST, and LDRGNT by totalling corresponding input variables over all 16 quarters in the planning horizon. Next, the routine checks if the total number of depot condemnations (LDCONT) exceeds the total number of depot reparable generations (LDRGNT). If so, LDCONT is reset to LDRGNT. Next, similar calculations are performed for each SRU. Finally, the routine checks if the total number of depot level SRU failure exposures exceed the number of SRU depot reparable generations. If so, the excess failures are assigned to a randomly selected quarter and program control returns to the calling program.

Entry point NOSET2(KQ) initializes counters used to simulate reparable generations for quarter KQ. The routine first sets the LRU counter variables LBCOND, LRTS, LDRGN, LNRTS, LBRGN, and LDCOND to the corresponding input variables associated with quarter KQ. Next, the routine similarly initializes the SRU counter variables NOSRUF, NBRTS, NNRTS, NBCOND, NDRGN, and NDCOND for each SRU. Program control then returns to the calling routine.

```

1  *SUBNAME=LINE/OBJ/NOSET1.0(BGD,NOGO)
2  *NOSET1,S
3  SUBROUTINE NOSET1
4  C
5  C      THIS ROUTINE INITIALIZES COUNTER ARRAYS THAT DETERMINE
6  C      THE NUMBER OF LRU AND SRU EVENTS TO BE GENERATED.
7  C      ENTRY POINT NOSET1 INITIALIZES DEPT EVENTS COUNTERS, WHILE
8  C      NOSET2(X0) INITIALIZES COUNTERS FOR QUARTER K0.
9  C      NOTE--THE LRU IS ITEM NUMBER N=1. ITEMS N=2,3, AND GREATER CORRESP
0  C      TO THE FIRST SRU, THE SECOND SRU, ETC.
1  COMMON/MENTRY/MENTRY
2  COMMON/IDBUG/IDBUG
3  COMMON/NBASES/NBASES
4  COMMON/NSRU/NSRU
5  COMMON/LBCOND/LBCOND
6  COMMON/LRTS/LRTS
7  COMMON/LNRTS/LNRTS
8  COMMON/LNRTST/LNRTST
9  COMMON/LDRGN/LDRGN
0  COMMON/LDRGN/LDRGN
1  COMMON/LDRGNT/LDRGNT
2  COMMON/LDCONB/LDCONB
3  COMMON/LOVGNT/LOVGNT
4  COMMON/LDCONT/LDCONT
5  COMMON/LOVCNT/LOVCNT
6  COMMON/LDREPT/LDREPT
7  COMMON/IDPA/IDPA(40)
8  COMMON/NOINL/NOINL(40)
9  COMMON/INRTS/INRTS(40)
0  COMMON/INRTS/INRTS(40)
1  COMMON/INRTST/INRTST(40)
2  COMMON/INBCOND/INBCOND(40)
3  COMMON/NOVCNT/NOVCNT(40)
4  COMMON/NOVRPT/NOVRPT(40)
5  COMMON/NOBRUP/NOBRUP(40)
6  COMMON/NDGRN/NDGRN(40)
7  COMMON/NDRGNT/NDRGNT(40)
8  COMMON/NDCONT/NDCONT(40)
9  COMMON/NDCOND/NDCOND(40)
0  COMMON/NOVGNT/NOVGNT(40)
1  COMMON/NDEXPO/NDEXPO(40)
2  COMMON/NINDGN/NINDGN(16,40)
3  COMMON/NOINDG/NOINDG(40)
4  C
5  COMMON/IBCOND/IBCOND(16,40)
6  COMMON/IBTS/IBTS(16,40)
7  COMMON/IDCOND/IDCOND(16,40)
8  COMMON/INRTS/INRTS(16,40)
9  COMMON/IDRGNT/IDRGNT(16,40)
0  COMMON/IOVCND/IOVCND(16,40)
1  C
2  COMMON/INQTR/INQTR

```

NOSET1

01 10-20-79 10.347 .5

```
3      COMMON/ITDAY/ITDAY
4      COMMON/ITQTR/ITQTR
5      COMMON/ITINV/ITINV
6      COMMON/NTIME/NTIME
7      COMMON/ITIME/ITIME
8      COMMON/IWT/IWT(20)
9      C
10     IF(IWT(3),EQ,1)WRITE(6,383)*****NOSET1 CALLED*
11     C
12     C          ASSIGN THESE FAILURES TO A RANDOMLY SELECTED QUARTER
13     C
14     C          COMPUTE TOTALS FOR THE PLANNING HORIZON(QUARTERS K=1,2,...IQPH):
15     C          (1)DEPOT LRU REPARABLE GENERATIONS
16     C          IQPH=16
17     C
18     C          LDCONT=0
19     C          LOVCNT=0
20     C          LOVGNT= 0
21     C          LNRTST=0
22     C          LDRGNT=0
23     C          DO 100 KKK=1,IQPH
24     C          K=KKK
25     C          LDCONT=LDCONT+IDCPND(K,1)
26     C          LOVCNT=LOVCNT+IOVEND(K,1)
27     C          LNRTST=LNRTST+INRTS(K,1)
28     C          LDRGNT=LDRGNT+IDRGH(K,1)
29     C          100 CONTINUE
30     C
31     C          LIMIT CONDEMNATIONS TO NO MORE THAN TOTAL GENERATIONS
32     C
33     C          IF(LDCONT.GT.LDRGNT)LDCONT=LDRGNT
34     C          LOVGNT=LNRTST+LDRGNT-LDCONT
35     C          IF(LOVCNT.GT.LOVGNT)LOVCNT=LOVGNT
36     C          LDREPT=LOVGNT-LOVCNT
37     C
38     C          (2)SRU DEPOT GENERATIONS
39     C
40     C          DO 400 JJJ=1,NSRU
41     C          J=JJJ
42     C          JJ=J+1
43     C          NDRGNT(J)=0
44     C          NDCONT(J)=0
45     C          NNRTST(J)=0
46     C          NOVGNT(J)=0
47     C          NOVCNT(J)=0
48     C          DO 200 KKK=1,IQPH
49     C          KQ=KKK
50     C          NNRTST(J)=NNRTST(J)+INRTS(KQ,JJ)
51     C          NDRGNT(J)=NDRGNT(J)+IDRGH(KQ,JJ)
52     C
53     C          SRU DEPOT CONDEMNATIONS
54     C
```

```

5      NDCONT(J)=NDCONT(J)+IDCOND(KO,JJ)
6      NOVCNT(J)=NOVCNT(J)+IOVCND(KO,JJ)
7      C
8      C
9      C
0      200 CONTINUE
1      C-----END OF QUARTER KO LOOP
2      C
3      C      LIMIT CONDEMNATIONS
4      C
5      C      IF(NDCONT(J).GT.NDRGNT(J))NDCONT(J)=NDRGNT(J)
6      C
7      NOVGNT(J)=NDRGNT(J)+NDRGNT(J)-NDCONT(J)
8      IF(NOVCNT(J).GT.NOVGNT(J))NOVCNT(J)=NOVGNT(J)
9      C
0      C
1      C      COMPUTE NO OF SRU DEPOT REPAIRS
2      C
3      NOBRPT(J)=NOVGNT(J)-NDRGNT(J)
4      C
5      C
6      C      SET TOTAL SRU DEPOT REPAIRS OVER HORIZON
7      C
8      NOBRPT(J)=NOVGNT(J)-NOVCNT(J)
9      C
0      C      (5)SRU FAILURE EXPOSURES
1
2      NDEXPO(J)=(NOVGNT+LDCONT)*IOPA(J)
3      C
4      C      (6)SRU INDEPENDENT FAILURES
5      C
6      NOIND=0
7      IF(NDEXPO(J).LT.NDRGNT(J))NOIND=NDRGNT(J)-NDEXPO(J)
8      C
9      C      ASSIGN THESE NOIND FAILURES TO A RANDOMLY SELECTED QUARTER
0      C
1      DO 300 KKK=1,IOPH
2          KQ=KKK
3          300 NINDGN(KQ,J)=0
4          IF(NOIND.LE.0)GO TO 320
5      C
6      DO 310 NP=1,NOIND
7          KQ=RANDU(2)*IOPH + 1
8          NINDGN(KQ,J)=NINDGN(KQ,J)+1
9      310 CONTINUE
0      C
1      320 CONTINUE
2      C
3      321 CONTINUE
4      IF(IWT(3).NE.1) GO TO 360
5      WRITE(6,323)J,(NINDGN(KQ,J),KQ=1,16)
6      323  FORMAT('---END, DEP REP GENS FOR SRU ',I4,' FOLLOW')

```

```

7      A      1614)
8      C
9      C
10     C      SET DEPOT COUNTERS
11     C
12     360 CONTINUE
13     IF(NODEPT(J).GE.0) GO TO 384
14     WRITE(6,380)J,NODEPT(J),ITIME
15     380 FORMAT(//T10,'*** WARNING ***1/'
16     &      T5,'NODEPT(',I2,')= ',I6,' WHICH IS RESET TO 0')
17     NODEPT(J)=C
18     381 CONTINUE
19     IF(INT(3).NE.1) GO TO 397
20     WRITE(6,383)" J LOVGNT NDRGNT NDCONT NDEXPO IQPA".
21     &      " NOVCNT NODEPT NOVGMT"
22     383 FORMAT(V)
23     WRITE(6,393) J,LOVGNT,NDRGNT(J),NDCONT(J),NDEXPO(J),IQPA(J),
24     &      NOVCNT(J),NODEPT(J),NOVGNT(J)
25     393 FORMAT(I3,I9,8I7)
26     397 CONTINUE
27     C
28     400 CONTINUE
29     C
30     C-----END OF SRU LOOP
31     C-----END OF NOSET1
32     IF(IWT(3).EQ.1)WRITE(6,383)"-----EXIT NOSET1"
33     C
34     RETURN
35     C
36     C      SET UP ARRAYS FOR QUARTER KQ
37     C
38     ENTRY NOSET2(KQ)
39     IF(IWT(3).EQ.1)WRITE(6,383)"***ENTER NOSET2"
40     C      GENERATE BASE LEVEL LRU EVENTS, AND ALL RELATED SRU EVENTS
41     C      COMPUTE APPLICABLE BASE GENERATIONS
42     C
43     LBCOND=IBCOND(KQ,1)
44     LRIS=IRIS(KQ,1)
45     LDRGN=IDRGN(KQ,1)
46     LNRTS=INRTS(KQ,1)
47     LBRGN=LRIS+LBCOND
48     LDCOND=IDCOND(KQ,1)
49     IF(LDCOND.GT.LDRGN)LDCOND=LDRGN
50     IF(IWT(3).EQ.1)WRITE(6,383)"LBRGN=",LBRGN," LRIS=",LRIS,
51     &      " LBCOND=",LBCOND," LNRTS=",LNRTS,
52     &      " LDRGN=",LDRGN," LDCOND=",LDCOND
53     C
54     C      FOR EACH SRU(K),K=1,2,...,NSRU
55     IF(IWT(3).EQ.1)
56     &      WRITE(6,383)" K NOSRUP NRRTS NNRTS NDCONT NOINDG".
57     &      " NDCONT NDRGN"
58     C

```

01 40-20-79 10,347 .5

DO 500 K=1, NSRU
KK=K+1

37

C

C

C

SET SRU BASE GENERATION DATA

NOSEUP(K)=IRTS(KQ, KK)+INRTS(KQ, KK)+IBCND(KQ, KK)

NBRTS(K)=IRTS(KQ, KK)

NBRTS(K)=INRTS(KQ, KK)

NBCND(K)=IBCND(KQ, KK)

C

C

C

C

C

SET SRU DEPOT GENERATION DATA

SET SRU DEPOT REP GENS ASSUMING THESE

GENERATIONS ARE INDEPENDENT OF LRU REP GENS.

C

C

NDRGN(K)=IDRGN(KQ, KK)

C

C

NDCOND(K)=IDCOND(KQ, KK)

IF(NDCOND(K), GT, NDRGN(K)) NDCOND(K)=NDRGN(K)

C

IF(INT(3), NE, 1) GO TO 487

WRITE(6, 493) K, NOSEUP(K), NBRTS(K), NBRTS(K), NBCND(K),

NOINDG(K), NDCOND(K), NDRGN(K)

A

493 FORMAT(I3, 2X, B17)

C

C

C

C

SET THE NUMBER OF LRU DEPOT REP GENS FOR THIS QUARTER.

AND THE NUMBER OF LRU DEPOT CONDEMNATIONS.

487 CONTINUE

500 CONTINUE

C

IF(INT(3), EQ, 1) WRITE(6, 583) "--EXIT NOSET2"

RETURN

END

Subroutine: RANBAS, RANBS1

Function:

These routines use Monte Carlo techniques to determine a randomly selected base.

Description:

Entry point RANBS1(IQTR) computes the cumulative probability of a failure at a given base K during quarter IQTR, and records the corresponding probabilities in the COMMON variable CPROB(K). This routine is called at the beginning of each simulated quarter. When a randomly selected base is needed, entry point RANBAS(NB) is called. This routine first obtains a uniform random number R in the range 0 to 1 by calling subroutine RANDU. It then does a table look-up using the base probability array CPROB to determine the base NB corresponding to the random number R. Program control then returns to the calling program.

```

1      SUBROUTINE RANBAS(NB)
2      C
3      C      MONTE CARLO TO DETERMINE REP GEN BASE LOCATION NB, WHERE
4      C      CPROB(KB) DENOTES THE CUMULATIVE PROBABILITY OF
5      C      A FAILURE AT BASE KB.
6      C      CPROB(K) MUST BE COMPUTED BY CALLING RANBS1 PRIOR TO
7      C      CALLING THIS ROUTINE.
8      C
9      COMMON/INT/INT(20)
10     COMMON/BN /BN(24,8)
11     COMMON/CPROB/CPROB(8)
12     COMMON/NBASES/NBASES
13     R = RANDU(22)
14     DO 20 KB = 1, NBASES
15     20 IF(R.LE.CPROB(KB)) GO TO 30
16     30 NB = KB
17     RETURN
18     C
19     ENTRY RANBS1(IQTR)
20     C      THIS ROUTINE COMPUTES THE CUMULATIVE PROBABILITY
21     C      OF A FAILURE AT A GIVEN BASE K.
22     C
23     C
24     IDBUG=INT(13)
25     C
26     C      COMPUTE TOTAL BASE FLYING HOURS THIS QUARTER
27     C
28     TBFH = 0.
29     DO 410 KB = 1, NBASES
30     TBFH = TBFH + BN(IQTR,KB)
31     410 CONTINUE
32     C
33     C      COMPUTE CUMULATIVE PROBABILITY OF A REP GEN AT BASE KB
34     C
35     C
36     TOTL = 0.
37     DO 420 KB = 1, NBASES
38     TOTL = TOTL + BN(IQTR,KB)
39     CPROB(KB)=TOTL/TBFH
40     420 CONTINUE
41     C
42     C      IF IDBUG.EQ.1, PRINT CPROB ARRAY.
43     C
44     C      IF(IDBUG.EQ.1)WRITE(6,423)NBASES,TBFH,
45     C      (K,BN(IQTR,K),CPROB(K),K=1,NBASES)
46     C      423 FORMAT(// "----RANBS1-- NBASES=",I3,
47     C      " TBFH=",F10.0,
48     C      8(/T40,"BASE=",I3," BN=",F8.0,
49     C      " CPROB=",F8.3))
50     C
51     RETURN
52     END

```

RANBAS

Subroutine: READFL, READF1Function:

This routine inputs D041 data describing the characteristics of individual LRU and SRUs. The routine is used by both the Events Generator and the Levels Computation System.

Description:

Entry point READF1 is called first to rewind Item Data file 07, and to read the first record on this file. Entry point READFL (IENDF) is then called to interpret the previously read record. The parameter IENDF is first set equal to 0. If an End-of- File is encountered during subsequent read operations, IENDF is set to 1 and program control returns to the calling program.

In READFL, the previously read data record is decoded to determine the record type. Logic then branches to the appropriate logic for decoding that specific type of input record. After decoding the record and setting the values of the corresponding COMMON variables, the routine reads the next record from file 07, and decodes it. If the record type of newly read record is different from that of previous record, logic jumps to statement 2000; otherwise logic again jumps to the decode logic described above.

When a new record is encountered, the routine checks (in lines 2120-2240) if the input data lies within "reasonable" ranges. If not, the associated variables are reset to standard values. Next, the routine uses the input data to set major time

variables used in the RIME model. These time variables include the base repair time, depot repair times, and various transportation times.

If the flag IOU equals 1, subroutine READFL outputs an item identification record to the Exogenous Event File (File 08). This option is not used when READFL is called by the Levels Computation System.

The above operations are continued until all input records for the current LRU/SRU family have been processed. After all records have been processed, if IOU equals 1, the trailer record of zeros is written to File 08. Finally, logic returns to the calling program.


```
1 *#RUN=JTIME/OBJ/READF2.0(BCD,NOGO)
2 *READF2,S
3 SUBROUTINE READFL(IENRT)
4 C-----BASEDA'S DATA
5 COMMON/NBASES/NBASES
6 COMMON/OSTDLT/OSTDLT(8)
7 C-----
8 COMMON/IOUT/IOUT
9 COMMON/NGROUP/NGROUP
10 COMMON/NFGRP /NFGRP
11 COMMON/IWT/IWT(1)
12 COMMON/NITEM/NITEM
13 C
14 COMMON/IBRGN/IBRGN(16,40)
15 COMMON/IBTS/IBTS(16,40)
16 COMMON/IBCOND/IBCOND(16,40)
17 COMMON/IBRTS/IBRTS(16,40)
18 COMMON/IDCOND/IDCOND(16,40)
19 COMMON/IDREP/IDREP(16,40)
20 COMMON/IOVCND/IOVCND(16,40)
21 COMMON/IPPORG/IPPORG(16,40)
22 COMMON/IDRGN/IDRGN(16,40)
23 COMMON/FSN/FSN(40)
24 COMMON/UCOST/UCOST(40)
25 COMMON/LTADM/LTADM(40)
26 COMMON/LTPROD/LTPROD(40)
27 COMMON/IBRT/IBRT(40)
28 COMMON/IBDTT/IBDTT(40)
29 COMMON/IDORT/IDORT(40)
30 COMMON/IDRT/IDRT(40)
31 COMMON/IOST/IOST(40)
32 COMMON/OIMDDR/OIMDDR(40)
33 COMMON/RTSF/RTSF(40)
34 COMMON/PDMF/PDMF(40)
35 COMMON/ROHF/ROHF(40)
36 COMMON/RMSTRF/RMSTRF(40)
37 COMMON/DDDDR/DDDDR(40)
38 COMMON/PJRCND/PJRCND(40)
39 COMMON/PNJRC/PNJRC(40)
40 COMMON/OIMBRB/OIMBRB(40)
41 COMMON/BCNF/BCNF(40)
42 COMMON/EJCNF/EJCNF(40)
43 COMMON/ENCNF/ENCNF(40)
44 COMMON/DOHCNF/DOHCNF(40)
45 COMMON/RMSCNF/RMSCNF(40)
46 COMMON/RMNR/ RMNR(40)
47 COMMON/RCOST/RCOST(40)
48 COMMON/IBPD/IBPD(40)
49 COMMON/IRINTR/IRINTR(40)
50 COMMON/IOPA/IOPA(40)
51 COMMON/SIMDDR/SIMDDR(40)
52 COMMON/IC1/IC1
```

READFL

```

03      COMMON/IC2/IC2
04      C-----+-----+-----+-----+-----+-----+
05      DIMENSION IDSHIP(8)
06      DIMENSION IZERO(17)
07      DATA IZERO/17*0/
08      C-----+-----+-----+-----+-----+
09      CHARACTER IZREC*100,IZREC*182,FSN*15,NAME*10,ERPC*1
10      CHARACTER WFSF*15
11      CHARACTER ALC*2
12      C
13      IENDF=0
14      IQDH=16
15      NITEM=0
16      N=1
17      C
18      WRITE(6,13)NGROUP,WFSF,IWT(1),IWT(2)
19      13   FORMAT(" -P-NGROUP=",I3,"  WFSF=",I3,"  IWT1=",
20              "I3,"  IWT2=",I3)
21      C
22      C
23      IPRINT=0
24      IF(NGROUP.GE.WFSF)IPRINT=IWT(1)
25      100 CONTINUE
26      C
27      C
28      C          WHAT TYPE OF RECORD HAVE WE READ?
29      C
30      GO TO(110,120,130,140,150,160,170,180,190,200,210),IREC
31      C
32      C          TYPE#1 RECORD:
33      C
34      110 DECODE(IZREC,113)ALC,FSN(N),NAME,UCCST(N),ERPC,
35      &    UI,LTADM(N),LTPROD(N),IBRT(N),IDRCB,BFCODE,SBCODE,
36      &    HPCODE,KOST(N),JOINSL,NJRSLS,IDFSL,CINDRR(N),SINHDDR(N),
37      &    RTSF(N),PDHF(N),BOHF(N),RMSTSF(N),CPDDR(N),BNRTSF,
38      &    PJRCND(N),PNJRC(N),OIMBR(N),BCONF(N),HJCNF(N),ENCNF(N),
39      &    DONCNF(N),RMSCNF(N),RHNR(N),NAPPL,MUSERS,IENDAT,
40      &    ISSN,RCOST(N),ISPHRS,JNSLD,ISPD(N),IRINTR(N),ISORN,
41      &    ISFLOW,ISRTIN,ISLT,IPRD,IOPA(N),IAF
42      113 FORMAT(A2,A45,A10,F9.2,A1,A2,I4,I2,2I8,A2,2A4,
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
02W 1282 INCOMPATIBLE W.D FIELD IN 'P' SPECIFICATION
03      &    3I2,I3,F5.4,A1,4F3.2,F5.4,F3.2,2F3.2,F5.4,6F3.2,I5,I3.
02W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE ,F5,4,
02W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE 4F3,2,
02W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE ,F5,4,
02W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE ,F3,2,

```

```

**W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE 2F3.2.
**W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE ,FS.4.
**W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE 6F3.2.
94      &          I4,A3,F9.2,I5,I2,6I3,I4,I4,I3)
95      JC1=IC1
96      JC2=IC2
97      JC3=IC3
98      JSBQ=ISEQ
99      IF(IPRINT,LT,1) GO TO 1000
100     WRITE(6,117) ISEQ,IC1,IC2,IREC,ALC,FSN(N),NAME,UCOST(N),ERRC,NGROUP
101     117 FORMAT(' SEQ# IC1 IC2 IREC ALC          FSN          '
102     & ' NOUN          COST          ERRC GRP',/I6,3I4,3X,A2,2X,A15.
103     &          A10,F10.2,7X,A1,I6//)
104     GO TO 1000
105     C
106     C          TYPE #2 RECORD
107     C
108     120 CONTINUE
109     C
110     C          TYPE #3 RECORD
111     C
112     130 DECODE(INREC,133) (IBRGN(K,N),K=1,ICPN)
113     133 FORMAT(2X,16I6)
114     IF(IPRINT,GE,2)CALL PBTPAS(IREC,N,IBRGN(1,N))
115     GO TO 1000
116     C
117     C          TYPE #4 RECORD
118     C
119     140 DECODE(INREC,133) (IRTS(K,N),K=1,ICPN)
120     IF(IPRINT,GE,2)CALL PBTPAS(IREC,N,IRTS(1,N))
121     GO TO 1000
122     C
123     C          TYPE #5 RECORD
124     C
125     150 DECODE(INREC,133) (IBCOND(K,N),K=1,ICPN)
126     IF(IPRINT,GE,2)CALL PBTPAS(IREC,N,IBCOND(1,N))
127     GO TO 1000
128     C
129     C          TYPE #6 RECORD
130     C
131     160 DECODE(INREC,133) (INRTS(K,N),K=1,ICPN)
132     IF(IPRINT,GE,2)CALL PBTPAS(IREC,N,INRTS(1,N))
133     GO TO 1000
134     C
135     C          TYPE #7 RECORD
136     C
137     170 DECODE(INREC,133) (IDRGN(K,N),K=1,ICPN)
138     IF(IPRINT,GE,2)CALL PBTPAS(IREC,N,IDRGN(1,N))
139     GO TO 1000
140     C
141     C          TYPE #8 RECORD
142     C

```

```

13      180 DECODE(INREC,133) (IDCOND(K,N),K=1,IQPH)
14      IF(IPRINT,GE,2)CALL PRTPAS(IREC,N,IDCOND(1,N))
15      GO TO 1000
16      C
17      C          TYPE #9 RECORD
18      C
19      190 DECODE(INREC,133) (IDREP(K,N),K=1,IQPM)
20      IF(IPRINT,GE,2)CALL PRTPAS(IREC,N,IDREP(1,N))
21      GO TO 1000
22      C
23      C          TYPE #10 RECORD
24      C
25      200 DECODE(INREC,133) (IOVCND(K,N),K=1,IQPH)
26      IF(IPRINT,GE,2)CALL PRTPAS(IREC,N,IOVCND(1,N))
27      GO TO 1000
28      C
29      C          TYPE #11 RECORD
30      C
31      210 DECODE(INREC,133) (IPPROG(K,N),K=1,IQPH)
32      IF(IPRINT,GE,2)CALL PRTPAS(IREC,N,IPPROG(1,N))
33      GO TO 1000
34      C
35      C          READ IN NEW RECORD OFF OF FILE #7
36      C
37      1000 CONTINUE
38      READ(7,230,END=3000)ISEQ,IC1,IC2,IREC,INREC
39      DECODE(INREC,233)ISNREC
40      233 FORMAT(A100)
41      230 FORMAT(I6,I1,I1,I2,A182)
42      C          TEST FOR CHANGES
43      IF((IOLDSQ.NE,ISEQ).AND.(IREC.NE,1))
44      &      WRITE(6,243)"WARNING=",
45      &      "IOLDSQ=ISEQ AND IREC=1"
46      243 FORMAT(V)
47      IF((IOLDSQ.NE,ISEQ).AND.(IREC.EQ,1))GO TC 2000
48      C
49      C
50      C          PROCESS THIS RECORD
51      C
52      GO TO 100
53      C
54      C          NEW RECORD IS AN 01
55      C
56      2000 CONTINUE
57      IF(IPRINT,LT,3) 30 TO 2090
58      WRITE(6,127)
59      &      UI,LTADN(N),LTPROD(N),IBRT(N),IDRCD,BPCODP,SMCODE,MPCODE
60      127 FORMAT(// " UI LTADN LTPROD IBRT IDRCD BPCODE SMCODE MPCODE",
61      &      /1X,A2,4I7,A7,A7)
62      WRITE(6,128)
63      &      IOST(N),IOINSL,NJRSI,IDFSL,OIMDER(N)

```

```

195      128 FORMAT(' IOST IOHSL NJRSL IDPSI OIMBDE' /
196      &          I6,I6,I7,I7,3X,F7.3 )
197      WRITE(6,129)
198      &          RTSF(N),PDNF(N),EONF(N),RMSTRF(N),BPDDR(N),BNPTSF
199      129 FORMAT(' RTSF PDNF EONF RMSTRF BPDDR BNPTSF' /
200      &          F7.2, F7.2, F7.2, F7.2, F6.0, F7.0 )
201      WRITE(6,139) PJRCND(N),PNJRC(N),OIMBBR(N),RCONF(N),
202      &          EJCNF(N),ENCNF(N)
203      139 FORMAT(' PJRCND PNJRC OIMBBR RCONF EJCNE ENCNF' /
204      &          6F7.2 )
205      WRITE(6,149)DOHCNF(N),RMSCNF(N),RMNRF(N),NAPPL,
206      &          NUSERS,IENDAT
207      149 FORMAT(' DOHCNF RMSCNF RMNRF NAPPL NUSERS IENDAT' /
208      &          3F7.2,2I6,2X,I7 )
209      WRITE(6,159)IESSEN,RCOST(N),IRPHRS,JRSLD,IBPD(N),IRINTP(N),
210      &          ISORN
211      159 FORMAT(' IESSEN RCOST IRPHRS JRSLD IBPD IRINTP ISORN' /
212      &          ,2X,A4,F10.2,I4,4I7 )
213      WRITE(6,169) ISFLOW,ISRTIN,IRLT,IPBD,IQPA(N),IAP
214      169 FORMAT(' ISFLOW ISRTIN IRLT IPBD IQPA IAP' /
215      &          6I7)
216      2090 CONTINUE
217      C
218      C          LIMIT INPUT PARAMETERS TO "REASONABLE" VALUES
219      C
220      IF(IOST(1).LE.0)IOST(1)=14
221      C
222      IF(IQPA(N).LT.1)IQPA(N)=1
223      IF(LTADM(N).LE.0)LTADM(N)=3
224      IF(LTPROD(N).LE.0)LTPROD(N)=9
225      IF(IBPD(N).LE.0)IBPD(N)=6
226      IF(ISFLOW.LE.0)ISFLOW=30
227      IF(IDRCD.LE.0)IDRCD=63
228      IF(IDRCD.LE.ISFLOW)IDRCD=ISFLOW+2
229      C
230      C          SET TIME PARAMETERS FOR ITEM N
231      C
232      IBRT(N)=IBPD(N)
233      IDRT(N)=ISFLOW
234      IBDTT(N)=IDRCD-ISFLOW
235      IDQRT(N)=ISFLOW
236      IBDORD=1
237      IOVSHR=IOST(1)-IBDORD
238      C
239      C          CONVERT PROCUREMENT LEADTIMES FROM MONTHS TO DAYS
240      C
241      LTADM(N)=30*LTADM(N)
242      LTPROD(N)=30*LTPROD(N)
243      C
244      C          SET ORDER AND SHIP TIME PARAMETERS
245      C
246      DO 2220 K=1,NBASES

```

```

17      JOST=IOST(1) + OSTDLT(K) + 0.5
18      IDSHIP(K)=JOST-IBDORD
19      IF(IDSHIP(N).LT.1)IDSHIP(N)=1
20      2220 CONTINUE
21      C
22      C
23      IC3=0
24      IF((INT(2).EQ.1).AND.(NGROUP.GE.NFGRP))
25      &      WRITE(6,2223)JC1,JC2,JC3,JSEQ,FSN(N),UCCST(N)
26      2223 FORMAT("SKU DATA=",B16,A15,F10.2)
27      IF((IOUT.EQ.1).AND.(NGROUP.GE.NFGRP))
28      &      WRITE(8)JC1,JC2,JC3,JSEQ,FSN(N),UCOST(N)
29      &      ,IOPA(N),IBDTT(N),IDBT(N),IBRT(N),IDORT(N),
30      &      ,LTADM(N),LTPROD(N),IBDORD,IOVSHP,IDSHIP
31      &      ,IOLDSQ,ISEQ
32      C
33      C      IF INT(2)>= 2, PRINT OUTPUT RECORD HEADEN
34      C
35      IF( (INT(2).GE.1).AND.(NGROUP.GE.NFGRP) )
36      &      WRITE(6,2023)JC1,JC2,JC3,JSEQ,
37      &      ,FSN(N),UCOST(N),IOPA(N),IBDTT(N),IDBT(N),IBRT(N),
38      &      ,IDORT(N),LTADM(N),LTPROD(N),IBDORD,IOVSHP,IDSHIP
39      2023 FORMAT(///" JC1 JC2 JC3 JSEQ      FSN      UCOST",
40      &      " IOPA IBDTT IDBT IBRT IDORT LTADM LTPROD",
41      &      " IBDORD IOVSHP"/
42      &      ,3I4,I5,A16,F10.2,9I7///" IDSHIP 1,2,..."/8I4)
43      IF(IPRINT.GT.0) WRITE(6,179)
44      179 FORMAT("0",50(1H,))//)
45      IF(IC1.NE.3) GO TO 300
46      N=N+1
47      NITEM=N
48      GO TO 100
49      C
50      C      IF IOUT=1, OUTPUT HEADER RECORD TO EXOGFILE
51      C      IF INT(2)=1, ALSO PRINT HEADER RECORD
52      300 CONTINUE
53      WFSN="EVENTS FOLLOW"
54      IF(NGROUP.LT.NFGRP) GO TO 320
55      IF(INT(2).EQ.1)WRITE(6,2223)0.0,0.0,IOLDSQ,WFSN,0.
56      IF(IOUT.EQ.1)WRITE(8)0.0,0.0,IOLDSQ,WFSN,0.,IZERO
57      320 CONTINUE
58      RETURN
59      C
60      C
61      C      READ END OF FILE 7
62      3000 CONTINUE
63      INDF=1
64      RETURN
65      C
66      ENTRY READF1
67      REWIND 07
68      READ(7,193,END=3000)ISEQ,IC1,IC2,IREC,INREC

```

04F 01 10-20-79 10,356 .S

48

299 DECODE(INREC,233)ISNREC
300 193 FORMAT(I6,I1,I1,I2,A182)
301 IOLDSQ=IS20
302 RETURN
303 END

***W 7 MEMORY EXPANDED; USE SLIMITS OR CORE= OPTION FOR NEXT RUN

Subroutine: SCHIND

Function:

This routine schedules independent SRU Events.

Calling Parameters:

K = The SRU involved.
NUMB = The number of events of the current type to be generated.
KTYPE = The event type to be generated.
KQTR = The quarter in which the events are to be generated.
KLOC = Event location code. If KLOC equals 0, depot rep gen events are scheduled; otherwise, the events are scheduled at a randomly selected base.

Description:

Subroutine SCHIND first establishes a Stock Keeping Unit number for this rep gen. If KLOC equals 0, the SKU is set equal to the Stock Keeping Unit number of the depot location for SRU K. Otherwise, the SKU is set equal to the Stock Keeping Unit number for SRU K at a randomly selected base. The reparable generation counter NJOB is then incremented, and a Monte Carlo process is employed to establish at a randomly selected time within quarter KQTR. Sub-

routine ENTER is then called to place the reparable generation event (Event Type 14) and an associated requisition for a serviceable replacement (Event Type 1) on the Future Event List. The program logic then branches depending on the type (KTYPE) of event to be generated.

If a base condemnation or base repair completion is to be scheduled, subroutine ENTER is called to place appropriate Type 15 or Type 18 events on the Future Events List. On the other hand, if a depot reparable generation is involved, a Monte Carlo process is used to determine if the asset is to be depot condemned. If so, a depot condemnation event (Event Type 15) is scheduled to occur at the depot location. Otherwise, a NRTS event (Event Type 19) is scheduled to simulate shipment of the asset to the depot. In addition, when a NRTS event is scheduled, a Monte Carlo process is used to determine if the NRTS asset is to be condemned at the depot, or if the asset will be reparable at that location. If the asset is to be condemned at the depot, an appropriate condemnation event (Event Type 15) is scheduled to occur at the depot after an appropriate transportation time. Otherwise, a repair completion event (Event Type 18) is scheduled to occur after a time delay which allows for the transportation of the asset back to the depot, and the repair of the asset at that location.

In all of the above cases, appropriate counters of SRU events are reset after each of the Monte Carlo calculations to ensure that the correct probability values are used in subsequent Monte Carlo computations.

T 01 10-20-79 13.849

```

1      SUBROUTINE SCHIND(K,NUMB,KTYPE,KQTR,KLOC)
2      C
3      C      THIS ROUTINE SCHEDULES NUMB INDEPENDENT KTYPE EVENTS
4      C      FOR SUB K TO OCCUR IN QUARTER KQTR.
5      C      IF KLOC=0, DEPOT REP GEN EVENTS ARE SCHEDULED
6      C      OTHERWISE, THE
7      C      EVENTS ARE SCHEDULED AT A RANDOMLY SELECTED BASE.
8      COMMON/NOVCNT/NOVCNT(40)
9      COMMON/NOVCNT/NOVCNT(40)
10     COMMON/NOVCNT/NOVCNT(40)
11     COMMON/NOVCNT/NOVCNT(40)
12     COMMON/NOVCNT/NOVCNT(40)
13     COMMON/NOVCNT/NOVCNT(40)
14     COMMON/NOVCNT/NOVCNT(40)
15     C
16     COMMON/IDRT/IDRT(40)
17     COMMON/IDRT/IDRT(40)
18     COMMON/IDRT/IDRT(40)
19     C
20     COMMON/IWT/IWT(20)
21     C
22     COMMON/NSRU/NSRU
23     COMMON/NSOB/NJOB
24     COMMON/ITIME/ITIME
25     COMMON/NBASES/NBASES
26     COMMON/ITDAY/ITDAY
27     COMMON/ITQTR/ITQTR
28     *****
29     C
30     IDBUG=INT(6)
31     C
32     MHHH=NUMB
33     DO 91 NNN=1,MHHH
34     C
35     C      ESTABLISH THE STOCKKEEPING NUMBER FOR THIS GEN GEN
36     C
37     IFNN=K+1
38     NSKU=(NBASES+2)*(K+1)
39     CALL NBASES(KBASE)
40     IF(KLOC.NE.0) NSKU=K*(NBASES+2) + 1 + KBASE
41     IF(IDBUG.EQ.1) WRITE(6,13)
42     C      *****SCHIND--K,NUMB,KTYPE,KQTR,KLOC,NSKU**
43     13 FORMAT(7)
44     IF(IDBUG.EQ.1) WRITE(6,23) K,NUMB,KTYPE,KQTR,KLOC,NSKU
45     23 FORMAT(10Z7)
46     C
47     C      INCREMENT NJOB
48     NJOB=NJOB+1
49     C
50     C      MONTE CARLO TO ESTABLISH EVENT TIME
51     C
52     ITIME=(KQTR-1)*ITQTR+RAND0(.2)*ITQTR

```

SCHIND

T 01 10-20-79 10.949

```

53      C
54      C          SCHEDULE REP GEN EVENT
55      C
56      JTINH=ITINH
57      CALL ENTER(JTINH,10,NSKU,9,NOOB)
58      C
59      C          SCHEDULE REQUISITION FOR REPLACEMENT UNITS
60      C
61      JTINH=ITINH+10
62      CALL ENTER(JTINH,1,NSKU,121)
63      C
64      C          BRANCH BY EVENT TYPE
65      C          DEPOT GENERATION=44
66      C          CONDEMNATION =15,NT0=18,NTS=19, OTHERS ARE PERIODS
67      C
68      IF(KTYPE,NO,14) GO TO 300
69      IF(KTYPE,NO,15) GO TO 100
70      IF(KTYPE,NO,18) GO TO 200
71      IF(KTYPE,NO,19) GO TO 400
72      PRINT, "***SCHIND--ILLEGAL EVENT TYPE*",KTYPE,K,
73      NUNS,KOTE,KLOC
74      STOP
75      C
76      C          CONDEMNATION EVENT
77      C
78      10 CONTINUE
79      JTINH=ITINH +10
80      CALL ENTER(JTINH, 15,NSKU,1,NJOB)
81      GO TO 500
82      C
83      C          NTS EVENT
84      C
85      20 CONTINUE
86      JTINH=ITINH +IBRT(IPSN)/VIWDAY
87      CALL ENTER(JTINH,18,NSKU,9,NOOB)
88      GO TO 500
89      C
90      C          MONTE CARLO TO DETERMINE IF SRU IS DEPOT CONDEMNED.
91      C
92      300 CONTINUE
93      DCONR=FLOAT(INDCOND(K))/FLOAT(INDRGN(K))
94      R=RANDU(1,2)
95      IF(INDRGN,NO,1)WRITE(6,13) ' <<<<<<SRU DEPOT REP GEN'.
96      '  ' INDCOND="INDCOND(K)."  INDRGN="INDRGN(K),
97      '  ' DCONR="DCONR,"  R="R
98      C
99      C          UPDATE DEPOT REP GEN COUNTS
00      C
01      INDRGN(K)=INDRGN(K)+1
02      INDRGT(K)=INDRGT(K)-1
03      C
04      IF(R.GT.DCONR) GO TO 400

```

T 01 10-20-79 13.349

```

05      C
06      C          SHU IS DEPOT CONDEMNED
07      C
08      NDCOND(R)=NDCOND(K)-1
09      NDCONT(R)=NDCONT(K)-1
10      JTIME=JTIME+10
11      CALL ENTER(JTIME,15,NSKU,1;WJOB)
12      GO TO 500
13      C
14      C          WRTS EVENT
15      C
16      40 CONTINUE
17      NOVcnt(R)=NOVcnt(K)-1
18      JTIME=JTIME+30
19      CALL ENTER(JTIME,19,NSKD,9,WJOB)
20      C
21      C          COMPUTE DEPOT STOCK KEEPING NUMBER, AND MONTE CARLO TO
22      C          DETERMINE IF THIS UNIT IS REPARABLE,
23      C
24      C
25      NSKDEP=(NBASNS+2)*K + 1
26      DCONR=FLOAT(NOVcnt(K))/FLOAT(NOVcnt(K)+NODRPT(K))
27      R=RANDU(0.8)
28      C
29      IF(IDBUS.NO.1)WRITE(6,13) " <<<< DEPOT LEVEL MAIN. = ",
30      &      "NOVcnt=",NOVcnt(K), " NODRPT=",NODRPT(K),
31      &      " DCONR=",DCONR, " R=",R
32      C
33      C          IF R IS LESS THAN OR EQUAL TO DCONR THIS IS CONDEMNED
34      C
35      IF(R.LE.DCONR)GO TO 490
36      C
37      C          WRTS IS REPARABLE. SCHEDULE REPAIR COMPLETION
38      C
39      NODRPT(R)=NODRPT(K)-1
40      JTIME=JTIME+(IDDT(IPSN)+IDRQ(IPSN))*ITDAY
41      CALL ENTER(JTIME,15,NSKDEP,1;WJOB)
42      GO TO 500
43      C
44      C          SHU IS CONDEMNED AT THE DEPOT
45      C
46      490 CONTINUE
47      NOVcnt(R)=NOVcnt(K)-1
48      JTIME=JTIME+(IDDT(IPSN)+IDRQ(IPSN))*ITDAY
49      CALL ENTER(JTIME,15,NSKDEP,1;WJOB)
50      NOVcnt(R)=NOVcnt(K)-1
51      C
52      500 CONTINUE
53      900 CONTINUE
54      C
55      IF(IDBUS.NO.1)WRITE(6,13) " <<<< WRTS SCHED "
56      C

```

T 01 10-20-79 10.949

57 RETURN
58 END

Subroutine: SRUEVT

Function:

This routine schedules all SRU events associated with the current LRU reparable generation.

Calling Parameters:

LRUREP = LRU Repairability Flag. LRUREP = 0 indicates the LRU is to be condemned, while ' RUREP = 1 indicates the LRU is reparable.

NSKLRU = The Stock Keeping Unit number for the LRU reparable generation.

Description:

For a detailed discussion of the functions of this routine, see the Event Generator Chapter of Volume I.

```

1  *SRUN=TIME/OBJ/SRUEVT,0(SCD,NOGO)
2  *SRUEVT,S GENERATE SRU EVENTS
3  SUBROUTINE SRUEVT(LRUREP,NSKLRLU)
4      C SCHEDULE SRU EVENTS ASSOCIATED WITH THE CURRENT LRU
5      C REP GEN, WHERE LRUREP=0 INDICATES LRU IS CONDEMNED.
6      C LRUREP=1 INDICATES LRU IS REPARABLE.
7      C
8      C NSKLRLU= STOCK KEEPING UNIT FOR LRU REP GEN.
9      C WHERE 1 DENOTES THE DEPOT.
10     C
11     COMMON/IWT/IWT(20)
12     COMMON/NJOB/NJOB
13     COMMON/NSRU/NSRU
14     COMMON/IOPA/IOPA(40)
15     COMMON/NOINL/NOINL(40)
16     COMMON/NBRTS/NBRTS(40)
17     COMMON/NDCONT/NDCONT(40)
18     COMMON/NDRGNT/NDRGNT(40)
19     COMMON/NNRTS/NNRTS(40)
20     COMMON/NBCOND/NBCOND(40)
21     COMMON/NOVCNT/NOVCNT(40)
22     COMMON/NOVGNT/NOVGNT(40)
23     COMMON/NOBRPT/NOBRPT(40)
24     COMMON/NOSRUF/NOSRUF(40)
25     C
26     COMMON/IDRT/IDRT(40)
27     COMMON/IDTT/IDTT(40)
28     COMMON/IBRT/IBRT(40)
29     C
30     COMMON/ITOTR/ITOTR
31     COMMON/ITDAY/ITDAY
32     COMMON/NBASES/NBASES
33     COMMON/ITINY/ITINY
34     COMMON/ITIME/ITIME
35     C
36     IF(IWT(6).EQ.1)WRITE(6,13)*****SRUEVT--LRUREP=,
37     LRUREP," NSKLRLU=",NSKLRLU
38     13 FORMAT(V)
39     C
40     C
41     C BEGIN SRU LOOP
42     C
43     C DO 400 K=1,NSRU
44     C
45     C SET INDEX FOR SRU TIME REFERENCES
46     C
47     C IFSM=K+1
48     C
49     C IF THERE ARE NO FAILURES OF THIS SRU, GO TO 400
50     C
51     C IF(NOINL(K).LE.0) GO TO 400
52

```

SRUEVT

01 10-20-79 10.322 .S GENERATE SRU EVENTS

57

```
03 C
04 C COMPUTE THE STOCK KEEPING NUMBER FOR SRU(K) AT BASE NB=
05 C
06 NSKSRU=(NBASES+2)*K+NSKLRU
07 C
08 C RECORD GENERATION OF NOINL(K) UNITS OF SRU K
09 C
10 JTIME=JTIME + 15
11 NOINLL=NOINL(K)
12 CALL ENTER(JTIME,14,NSKSRU,NOINLL,NJOB)
13 C
14 C
15 C IF LRU IS TO BE CONDEMNED(LRUREP=0),GO TO 160.
16 C
17 C IF(LRUREP.LE.0) GO TO 160
18 C
19 C SCHEDULE A REQUISITION TO REPLACE THE NOINL(K) UNITS.
20 C
21 JTIME=JTIME +20
22 LPRI=(100*NJOB+2)
23 CALL ENTER(JTIME,1,NSKSRU,NOINLL,LPRI)
24 GO TO 190
25 C
26 C SINCE LRU IS CONDEMNED, SCHEDULE SERVICEABLE RETURNS
27 C (EVT CODE 4). IF ANY. THEN GO TO 190.
28 C
29 160 CONTINUE
30 NS=IQPA(K)-NOINL(K)
31 JTIME=JTIME+20
32 IF(NS.GT.0) CALL ENTER(JTIME,4,NSKSRU,NS,NJOB)
33 C
34 190 CONTINUE
35 C
36 C IF THERE ARE NO SRU K EVENTS, GO TO 310
37 C IF(NOINL(K).LE.0) GO TO 310
38 C
39 C
40 C
41 C FOR EACH RELATED SRU FAILURE, LOOP FOR KK=1,2,....
42 C NOINL(K).
43 C
44 DO 300KK=1,NOINL(K)
45 C
46 C IF THIS IS A BASE LRU REP GEN. GO TO 240
47 C
48 NSKOVH=NBASES+2
49 IF( (NSKLRU.GT.1) .AND. (NSKLRU.LT.NSKOVH) ) GO TO 240
50 C
51 C REPARABLE LRU IS AT DEPOT
52 C DECREMENT COUNT OF OVERHAUL GENERATIONS
53 NOVONT(K)=NOVONT(K)-1
54 C
```



```

15 C
16 C MONTE CARLO TO DETERMINE IF FAILURE KK IS DEPOT
17 C CONDEMNED.
18 C
19 DCORR=FLOAT(NDCONT(K))/FLOAT(NDRGNT(K))
20 R=RANDU(0.8)
21 C
22 IF(IWT(6).EQ.1)WRITE(6,13) UNIT="KK," NDCONT="NDCONT(K),
23 " NDRGNT="NDRGNT(K)," DCORR="DCORR," R="R
24 C
25 IF(R.LE.DCORR)GO TO 230
26 C
27 C SCHEDULE SRU REPAIR COMPLETION
28 C
29 JTIME=JTIME+TRET(IPR)*TTRAY
30 CALL ENTER(JTIME,18,NKSRU.1,NJOB)
31 NDRGNT(K)=NDRGNT(K)-1
32 GO TO 300
33 C
34 C SCHEDULE SRU CONDEMNATION
35 C
36 230 CONTINUE
37 JTIME=JTIME + 30
38 CALL ENTER(JTIME,15,NKSRU.1,NJOB)
39 NDCONT(K)=NDCONT(K)+1
40 NDRGNT(K)=NDRGNT(K)-1
41 GO TO 300
42 C
43 C MONTE CARLO TO DETERMINE IF FAILURE KK IS RTS. NRTS O
44 C ORCONDEMNED,
45 C
46 240 CONTINUE
47 PRTS=FLOAT(NRTS(K))/FLOAT(NRTS(K)+NNRTS(K)+NDCOND(K))
48 CRTS=FLOAT(NDCOND(K))/FLOAT(NRTS(K)+NNRTS(K)+NDCOND(K))+PRTS
49 R=RANDU(0.7)
50 IF(IWT(6).EQ.1)WRITE(6,313)KK,PRTS,CRTS,R
51 313 FORMAT(" UNIT="X5," PRTS="F5.3," CRTS="F5.3," R="F5.3)
52 C
53 IF R IS <= PRTS, FAILURE IS AN RTS.
54 C
55 IF(R.LE.PRTS) GO TO 250
56 C
57 IF R IS LESS THAN OR EQUAL TO CRTS, FAILURE IS CONDEMNATION.
58 C
59 IF (R.LE.CRTS)GO TO 270
60 C
61 IF R IS GREATER THAN CRTS, FAILURE IS A NRTS.
62 C SCHEDULE NRTS EVENT. DECREMENT NRTS COUNTER.
63 C
64 JTIME=JTIME+30
65 CALL ENTER(JTIME,19,NKSRU.1,NJOB)
66 NNRTS(K)=NNRTS(K)-1

```

```

7      GO TO 280
8      C
9      C      SCHEDULE RTS EVENT, AND DECREMENT RTS COUNTER,
10     C
11     250 JTIME=ITIME+IBRT(IPSN)*ITDAY
12     CALL ENTER(JTIME,18,NSKSRU,1,NJOB)
13     NRTS(K)=NRTS(K)-1
14     GO TO 300
15     C
16     C      SCHEDULE COND EVENT, AND DECREMENT COND COUNTER,
17     C
18     270 CONTINUE
19     JTIME=ITIME + 30
20     CALL ENTER(JTIME,15,NSKSRU,1,NJOB)
21     NBCOND(K)=NBCOND(K)-1
22     GO TO 300
23     C
24     C      IF FAILURE IS NRTS, MONTE CARLO TO DETERMINE IF THE SRU
25     C      IS TO BE REPAIRED OR CONDEMNED AT THE DEPOT. THEN SCHEDULE
26     C      THE CORRESPONDING REPAIR COMPLETION OR CONDEMNATION
27     C      EVENT.
28     C
29     280 CONTINUE
30     C      DECREMENT COUNT OF OVERHAUL GENERATIONS
31     C
32     NOVCNT(K)=NOVCNT(K)-1
33     NSKDEP=(NBASES+2)*K + 1
34     DCONR=FLOAT(NOVCNT(K))/FLOAT(NOVCNT(K)+NSKDEPT(K))
35     R=RANDU(0,8)
36     C
37     C      IF R IS LESS THAN OR EQUAL TO DCONR, NRTS IS CONDEMNED
38     C
39     IF(R.LE.DCONR)GO TO 290
40     C
41     C      NRTS IS REPARABLE. SCHEDULE REPAIR COMPLETION
42     C
43     IPSN=K+1
44     JTIME=ITIME+(IBDRT(IPSN)+IDRT(IPSN))*ITDAY
45     CALL ENTER(JTIME,18,NSKDEP,1,NJOB)
46     NOBRPT(K)=NOBRPT(K)+1
47     GO TO 300
48     C
49     C      SRU IS CONDEMNED AT THE DEPOT
50     C
51     290 CONTINUE
52     IPSN=K+1
53     JTIME=ITIME + (IBDRT(IPSN)*ITDAY)
54     CALL ENTER(JTIME,15,NSKDEP,1,NJOB)
55     NOVCNT(K)=NOVCNT(K)-1
56     C
57     C      END OF KK FAILURE LOOP
58     C

```

F 01 10-20-79 10.322 ,S GENERATE SRU EVENTS

```
19      300 CONTINUE                                     60
10      310 CONTINUE
11      C
12      C          END OF SRU(K) LOOP
13      C
14      400 CONTINUE
15      500 CONTINUE
16      IF(IWT(6).EQ.1) WRITE(6,13)'-----EXIT SRUEVT'
17      RETURN
18      END
```

Subroutine: SRUIND**Function:**

This routine scans the SRU event counter arrays, and schedules independent SRU events as needed.

Description:

This routine scans the events counter arrays for each SRU. The routine is called after all LRU events have been generated. If any of the SRU arrays are non-zero, the remaining SRU events are assumed to occur independently of LRU reparable generations. Consequently, subroutine SRUIND places appropriate events on the Future Events List to simulate these independent SRU events.

If the counter of SRU base RTS events (NBRTS(K)) for SRU K is greater than zero, subroutine SCHIND is called to place NBRTS(K) type 18 events on the Future Events List for SRU K. Similarly, if the SRU event counters NBCOND(K), NNRTS(K), or NDRGN(K) is greater than zero for any SRU K, subroutine SCHIND is called to schedule the associated type 15, 19, and 14 events, respectively.

```

1      SUBROUTINE SRUPRT
2      COMMON/NSRU/NSRU
3      COMMON/IOPA/IOPA(40)
4      COMMON/NOINL/NOINL(40)
5      COMMON/NBRTS/NBRTS(40)
6      COMMON/NNRTS/NNRTS(40)
7      COMMON/NBCOND/NBCOND(40)
8      COMMON/NOVCNT/NOVCNT(40)
9      COMMON/NODRPT/NODRPT(40)
10     COMMON/NOSRUF/NOSRUF(40)
11     COMMON/NDRGN/NDRGN(40)
12     COMMON/NDRGNT/NDRGNT(40)
13     COMMON/NDCOND/NDCOND(40)
14     COMMON/NDCONT/NDCONT(40)
15     COMMON/NOVGNT/NOVGNT(40)
16     COMMON/NDEXPO/NDEXPO(40)
17     WRITE(6,413)
18     413 FORMAT( 'NJ',T10,' IOPA',T20,' NOINL',T30,
19             ' NBRTS',T40,' NNRTS',T50,' NBCOND',
20             T60,' NOVCNT',T70,' NODRPT',T80,' NOSRUF')
21     DO 431 NJ=1,NSRU
22     WRITE(6,423) NJ,IOPA(NJ),NOINL(NJ),NBRTS(NJ),
23             NNRTS(NJ),NBCOND(NJ),NOVCNT(NJ),
24             NODRPT(NJ),NOSRUF(NJ)
25     423 FORMAT ( I3,T6,8I10)
26     431 CONTINUE
27     WRITE(6,433)
28     433 FORMAT( 'NJ',T10,' NDRGNT',T20,' NDCONT',T30,
29             ' NOVGNT',T40,' NDEXPO',T50,' NDRGN',
30             T60,' NDCOND')
31     DO 437 NJ=1,NSRU
32     WRITE(6,436) NJ,NDRGNT(NJ),NDCONT(NJ),NOVGNT(NJ),
33             NDEXPO(NJ),NDRGN(NJ),NDCOND(NJ)
34     436 FORMAT ( I3,T6,10I10)
35     437 CONTINUE
36     RETURN
37     END

```

2T 01 10-20-79 10.345 LS

```

1  *SRUN=LTIME/655/SRUIND.0(NCD,NOSOT
2  *SRUIND.5
3  SUBROUTINE SRUIND
4      C      SCAN SRU EVENT COUINTER ARRAYS, AND SCHEDULE INDEPENDENT
5      C      SRU EVENTS AS NEEDED.
6      C
7      COMMON/NSRU/NSRU
8      COMMON/EBRTS/EBRTS(40)
9      COMMON/NSCOND/NSCOND(40)
10     COMMON/ENRTS/ENRTS(40)
11     COMMON/NDRGW/NDRGW(40)
12     COMMON/ITINV/ITINV
13     C
14     C
15     DO 900 KK=1,NSRU
16     K=KK
17     C
18     C      SCHEDULE RTS EVENTS (CODE 18)?
19     C
20     KQTR=ITINV
21     IF(ENRTS(K).GT.0)CALL SCHIND(K,ENRTS(K),18,KQTR,1)
22     C
23     C      SCHEDULE CONDENSATION EVENT (CODE 15)?
24     C
25     IF(NSCOND(K).GT.0)CALL SCHIND(K,NSCOND(K),15,KQTR,1)
26     C
27     C      SCHEDULE WRTS EVENT A (CODE 19)?
28     C
29     IF(WRTS(K).GT.0)CALL SCHIND(K,WRTS(K),19,KQTR,1)
30     C
31     C      SCHEDULE DEFOT RES WENS, IF ANY
32     C
33     IF(NDRGW(K).GT.0)CALL SCHIND(K,NDRGW(K),18,KQTR,0)
34     900 CONTINUE
35     RETURN
36     END

```

SRUIND

T 01 11-07-79 14,524

```

1      SUBROUTINE SRUPRT
2      COMMON/NSRU/NSRU
3      COMMON/IQPA/IQPA(40)
4      COMMON/NOINL/NOINL(40)
5      COMMON/NBRTS/NBRTS(40)
6      COMMON/NBRTS/NBRTS(40)
7      COMMON/NBCOND/NBCOND(40)
8      COMMON/NOVCNT/NOVCNT(40)
9      COMMON/NOVRPT/NOVRPT(40)
10     COMMON/NOSRUF/NOSRUF(40)
11     COMMON/NDRGN/NDRGN(40)
12     COMMON/NDRGNT/NDRGNT(40)
13     COMMON/NDCOND/NDCOND(40)
14     COMMON/NDCONT/NDCONT(40)
15     COMMON/NOVGNT/NOVGNT(40)
16     COMMON/NDEXPO/NDEXPO(40)
17     WRITE(6,413)
18     413 FORMAT( 'NJ',T10,' IQPA',T20,' NOINL',T30,
19     &          ' NBRTS',T40,' NBRTS',T50,' NBCOND',
20     &          T60,' NOVCNT',T70,' NOVRPT',T80,' NOSRUF')
21     DO 430 NJ=1,NSRU
22     WRITE(6,423) NJ,IQPA(NJ),NOINL(NJ),NBRTS(NJ),
23     &          NBRTS(NJ),NBCOND(NJ),NOVCNT(NJ),
24     &          NOVRPT(NJ),NOSRUF(NJ)
25     423 FORMAT ( I3,T6,8I10)
26     430 CONTINUE
27     WRITE(6,433)
28     433 FORMAT( 'NJ',T10,' NDRGNT',T20,' NDCONT',T30,
29     &          ' NOVGNT',T40,' NDEXPO',T50,' NDRGN',
30     &          T60,' NDCOND')
31     DO 437 NJ=1,NSRU
32     WRITE(6,436) NJ,NDRGNT(NJ),NDCONT(NJ),NOVGNT(NJ),
33     &          NDEXPO(NJ),NDRGN(NJ),NDCOND(NJ)
34     436 FORMAT ( I3,T6,10I10)
35     437 CONTINUE
36     RETURN
37     END

```

SRUPRT

REFERENCES

1. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluator, Volume I: Model Description, TR-80-01, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 153 pp.
2. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluation Volume II, Program Listings and Narratives, TR-80-02, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 298 pp.
3. Demmy, W. Steven, An Empirical Evaluation of Proposed Stockage Policies for Recoverable Item Management, TR-80-03, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 173 pp.
4. Demmy, W. Steven and Victor J. Presutti, Jr., Multi-Echelon Inventory Theory in the Air Force Logistics Command, Working Paper 76-3011-27, Department of Management, Wright State University, Dayton, Ohio 45435.
5. Muckstadt, John A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System", Management Science, v20, n4, December, 1973, pp. 472-481.